Workshop Notes of the Seventh International Workshop "What can FCA do for Artificial Intelligence?"

Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph

Workshop Notes

Seventh International Workshop

"What can FCA do for Artificial Intelligence?"

FCA4AI 2019

International Joint Conference on Artificial Intelligence

IJCAI 2019

August 10 2019

Macao, China

Editors
Sergei O. Kuznetsov (NRU HSE Moscow)
Amedeo Napoli (LORIA Nancy)
Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2019/

Preface

The six preceding editions of the FCA4AI Workshop showed that many researchers working in Artificial Intelligence are deeply interested by a well-founded method for classification and mining such as Formal Concept Analysis (see `http://www.fca4ai.hse.ru/`). FCA4AI was co-located with ECAI 2012 (Montpellier), IJCAI 2013 (Beijing), ECAI 2014 (Prague), IJCAI 2015 (Buenos Aires), ECAI 2016 (The Hague), and finally with IJCAI/ECAI 2018 (Stockholm). All the proceedings of the preceding editions are published as CEUR Proceedings (`http://ceur-ws.org/Vol-939/`, `http://ceur-ws.org/Vol-1058/`, `http://ceur-ws.org/Vol-1257/`, and `http://ceur-ws.org/Vol-1430/`, `http://ceur-ws.org/Vol-1703/`, and `http://ceur-ws.org/Vol-2149/`). This year, the workshop has again attracted researchers working on actual and important topics related to FCA, showing the diversity and the richness of the relations between FCA and AI.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies (implications) which can be used for many Artificial Intelligence needs, e.g. knowledge discovery, learning, knowledge representation, reasoning, ontology engineering, as well as information retrieval and text processing. Recent years have been witnessing increased scientific activity around FCA, in particular a strand of work emerged that is aimed at extending the possibilities of FCA w.r.t. knowledge processing, such as work on pattern structures and relational context analysis. These extensions are aimed at allowing FCA to deal with more complex data, both from the data analysis and knowledge discovery points of view. Then these investigations provide new possibilities for AI practitioners in the framework of FCA. Accordingly, we are interested and discuss the following issues at FCA4AI:

- How can FCA support AI activities such as knowledge processing (knowledge discovery, knowledge representation and reasoning), learning (clustering, pattern and data mining), natural language processing, and information retrieval.
- How can FCA be extended in order to help Artificial Intelligence researchers to solve new and complex problems in their domains.

In addition, the 3rd workshop on "Formal Concept Analysis for Knowledge Discovery" (FCA4KD 2019) was held at the Faculty of Computer Science of National Research University Higher School of Economics (NRU HSE, Moscow, Russia) on June 7, 2019. FCA4KD is an event which is close to FCA4AI, as the goal of the FCA4KD is to attract researchers applying FCA-based methods of knowledge discovery in various subject domains. There was an invited talk by Andrey Rodin on the problem of justification of knowledge discovery. In addition, there were 6 regular contributions, three of which were selected for the current volume. Sergei O. Kuznetsov would like to acknowledge the support of the NRU HSE University Basic Research Program funded by the Russian Academic Excellence Project 5-100.

The Workshop Chairs

Sergei O. Kuznetsov
National Research University Higher School of Economics, Moscow, Russia

Amedeo Napoli
Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France

Sebastian Rudolph
Technische Universität Dresden, Germany

Program Committee

Jaume Baixeries (UPC Barcelona, Catalunya),

Aleksey Buzmakov (National Research University HSE Perm, Russia),

Victor Codocedo (UFTSM Santiago de Chile, Chile),

Elizaveta Goncharova (NRU Higher School of Economics, Moscow, Russia),

Marianne Huchard (LIRMM/Université de Montpellier, France),

Dmitry I. Ignatov (National Research University HSE Moscow, Moscow, Russia),

Sergei Kuznetsov (National Research University HSE Moscow, Russia),

Mehdi Kaytoue (INSA-LIRIS Lyon, France),

Florence Le Ber (ENGEES/Université de Strasbourg, France),

Amedeo Napoli (Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France),

Sergei A. Obiedkov (NRU Higher School of Economics, Moscow, Russia),

Sebastian Rudolph (Technische Universität Dresden, Germany),

Dmitry Vinogradov (Russian Academy of Science, Moscow, Russia).

Contents

Enabling natural language analytics over relational data using Formal Concept Analysis

C. Anantaram, Mouli Rastogi, Mrinal Rawat, and Pratik Saini

TCS Research, Tata Consultancy Services Ltd, Gwal Pahari, Gurgaon, India
(c.anantaram; mouli.r; rawat.mrinal; pratik.saini) @tcs.com

Abstract. Analysts like to pose a variety of questions over large relational databases containing data on the domain that they are analyzing. Enabling natural language question answering over such data for analysts requires mechanisms to extract exceptions in data, find steps to transform data, detect implications in the data, and apply classifications on the data. Motivated by this problem, we propose a semantically enriched deep learning pipeline that supports natural language question answering over relational databases and uses Formal Concept Analysis to find exceptions, classification and transformation steps. Our framework is based on a set of deep learning sequence tagging networks which extracts information from the NL sentence and constructs an equivalent intermediate sketch, and then maps it into the actual tables and columns of the database. The output data of the query is converted into a lattice structure which results into the (extent,intent) tuples. These tuples are then analyzed to find the exceptions, classification and transformation steps.

1 Introduction

Data analysts have to deal with a large number of complex and nested queries to dig out hidden insights from the relational datasets, spread over multiple files. Extraction of the relevant result corresponding to a given query can be easily done through a deep learnt NLQA framework, but to detect further explanations, facts, analysis and visualizations from queried output is a challenging problem. This kind of data analysis over query's result can be handled by Formal Concept Analysis, a mathematical tool that results in a concept hierarchy, makes semantical relations during the queries, and also can find the implications as well as asociations in the given dataset, can unify data and knowledge and is capable of information engineering as well as data mining. So for enabling NL analytics over such datasets for analysts, we present in this paper, a semantically enriched deep learning pipeline that a) enables natural language question answering over relational databases using a set of deep learnt sequence tagging networks, and b) carries out regularity analysis over the query results using Formal Concept Analysis to interactively explore, discover and analyze the hidden structure in the selected data [12] [11]. The deep learnt sequence tagging pipeline extracts information from the NL sentence and constructs an equivalent intermediate

sketch, and then uses that sketch to formulate the actual database query on the relevant tables and columns. Query results are used in Formal Concept Analysis to create a lattice structure of the objects and attributes. The obtained lattice structure is then used to find exceptions in the data, classification of a new object and also to find the set of steps to transform the data from one structure to another structure.

2 Formal Concept Analysis

Formal Concept Analysis provides a theoretical framework for learning hierarchies of knowledge clusters called formal concepts. A basic notion in FCA is the formal context. Given a set G of objects and a set M of attributes (also called properties), a formal context consists of a triple (G, M, I) where I specifies (Boolean) relationships between objects of G and attributes of M , i.e., I $\subseteq$ G $\times$ M .Usually, formal contexts are given under the form of a table that formalizes these relationships. A table entry indicates whether an object has the attribute, or not. Let $I(g) = \{m \in M; (g, m) \in I\}$ be the set of attributes satisfied by object g , and let $I(m) = \{g \in G; (g, m) \in I\}$ be the set of objects that satisfy the attribute m . Given a formal context (G, M, I) . Two operators $()'$ define a Galois connection between the powersets (P(G),$\subseteq$) and (P(M),$\subseteq$), with A$\subseteq$G and B$\subseteq$M:

$$A' = \{m \in M | \forall g \in A : gIm\}$$

and

$$B' = \{g \in G | \forall m \in B : gIm\}$$

.
That is to say, A$'$ is the set of all attributes which is satisfied all objects in A , whereas B$'$ is the set of all objects which satisfies all attributes in B . A formal concept of (G,M,I) is defined as a pair (A,B) with A$\in$G , B$\in$ M , A$'$=B and B$'$=A. A is called the extent of the formal concept (A,B), whereas B is called the intent.The set of all formal concepts of (G, M, I) equipped with a subconcept-superconcept partial order $\leq$ is the concept lattice denoted by $\mathcal{L}$. The and is defined as:
For A_1,$A_2\subseteq$ G and B_1,$B_2\subseteq$ M

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2 (equivalent to B_2 \subseteq B_1)$$

In this case, the concept (A_1, B_1) is called sub-concept and the concept (A_2, B_2) is called super-concept.

2.1 Association and Implication Rules

Given a formal context (G,M,I) there are extracted exact rules and approximate rules (rules with statistical values, for example, support and confidence).

These rules express in an alternative way the underlying knowledge. These rules are significant as they expresses the underlying knowledge of interaction among attributes.The exact rules are classified as implication rules while the approximation rules are classified as association rules.
Definition Given a formal context whose attributes set is M. An implication is an expression S $\Longrightarrow$ T, where S,T $\subseteq$ M. An implication S $\Longrightarrow$ T, extracted from a formal context, or respective concept lattice, have to be such that S$'$ $\subseteq$ T$'$. In other words: every object which has the attributes of S, also have the attributes of T. If X is a set of attributes, then X respects an implication S $\Longrightarrow$ T iff S $\not\subseteq$ X or T $\not\subseteq$ X. An implication S $\Longrightarrow$ T holds in a set $\{X_1, ..., X_n\} \subseteq$ M iff each X_i respects S $\Longrightarrow$ T.
Definition Given a threshold minsupp $\in$ [0, 1], where the support

$$supp(X) := \frac{card(X')}{card(G)}(withX' := g \in G|\forall m \in X : (g, m) \in I),$$

association rules are determined by mining all pairs X $\Longrightarrow$ Y of subsets of M such that

$$supp(X \implies Y) := supp(X)$$

is above the threshold minsupp, and the confidence

$$conf(X \implies Y) := \frac{supp(X \cup Y)}{supp(X)}$$

is above a given threshold minconf $\in$ [0, 1].

3 Methodology

We present a novel approach where a natural language sentence is converted into the sketch (Listing 1.1) which uses deep learning models and then further using the sketch to construct the database query (SQL) and fetch the output. This output is then taken to derive some explanations or interesting facts, find outliers or exceptions and rationalize the queried data if required (fig:1).
In order to generate the query sketch, we have a pipeline of multiple sequence tagging deep neural networks: Predicate Finder Model (Select Clause), Entity Finder Model (Values in Where Clause), Meta Type Model, Operators and Aggregation Model (all using bi-directional LSTM network along with a CRF (conditional random field) output layer), where the natural language sentence is processed as a sequence tagging problem.
The architecture uses an ELMO embedding that are computed on top of two-layer bidirectional language models with character convolutions as a linear function of the internal network states [16]. Also the character-level embedding is used as it has been found helpful for specific tasks and to handle the out-of-vocabulary problem. The character-level representation is then concatenated with a word-level representation and feed into the bi-directional LSTM as input. In the next step, a CRF Layer yielding the final predictions for every word is

used [8]. We have Z = $(z_1; z_2; ...; z_n)$ as the input sentence and P to be the scores output by Bi-LSTM network. $Q_{i,j}$ is the score of a transition from *tag* i to *tag* j for the sequence of predictions $Y = (y_1; y_2; ...; y_n)$. Finally the score is defined as :

$$s(Z; Y) = \sum_{i=0}^{n} Q_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

Models details

To generate the query sketch we use four different models using the same architecture (BiLSTM-CRF) [17] explained above, where the natural language sentence is processed as a sequence tagging problem. The neural network then predicts the tag for each word using which predicates, entities, and values in the sentence are identified, and an intermediate Sketch (independent of underlying database) is created. The Sketch is then mapped into the columns of the tables with conditions to construct the actual SQL query. In the sketch generation process the order of the models matters as the input of the next model depends on the output of previous model. To train the models, we had to create the annotations. In the cases where predicate/entities present in the sentence got the direct match with columns or values present in the actual database, we extracted them using a script and in the rest of the cases we have manually annotated the data.

- **Predicate Finder Model(Select Clause):** This model identifies the target concepts (predicates) from the NL sentence. In case of database query language, predicate refers to the SELECT part of the query. Once predicates are identified, it becomes easier to extract entities from the remaining sentence.
- **Entity Finder Model(Values in Where Clause):** This model identifies the relations(values/entities) in the query. In some cases the model misses/-capture some words. To tackle this issue predicted value in the `Apache-Solr` is searched. The structured data for the domain is assumed to be present in Lucene. After the search we picked the entity from the database which has the highest similarity score.
- **Meta Type Model:** This model identifies the type of concepts (predicates and values) at the node or table level. If a concept is present in more than one table, type information helps in the process of disambiguation. This helps in making the overall framework domain agnostic.
- **Aggregations and Operators Model:** In this model, aggregations and operators are predicted for predicates and entities respectively. Our framework currently supports following set of aggregation functions: count, groupby, min, max, sum, asc sort, desc sort. Similarly, following set of operators are also supported: $=;>;<;<>;\geq;\leq;$*like*.

The models are trained independently and do not share any internal representations. However, the input of one model depends on the previous. For example, once predicates are identified we replace the predicate part in the NL sentence with some token before passing it to the next model. We capture this information from the NL sentence and create an intermediate representation (Sketch)

which is further passed to the query generator(neo4j knowledge graphs), to construct the SQL or another database query and yields results. Result table of the query is then converted into its equivalent formal context, which is a triplet of objects, attributes and incidence relation between them. This formal context is used to extract the implication and association rules [10] and create a concept lattice which derives all possible formal concepts from the context and orders them according to a subconcept-superconcept relationship [15]. This conceptual hierarchy of the queried output is further used for knowledge discovery that is implicitly present in it. Here we are focusing on three types of analysis over queried data from a relational database.

Listing 1.1: Sketch

```
{
"select":
[
{
"pred_hint": model
},
{
"pred_hint": horsepower,
"aggregation": desc_sort,
}]
"conditions":
{
"pred_hint": cylinders,
"value": 4,
"operator": =
}
}
```

3.1 Outliers Analysis

This is first type of analysis that could be perform in the queried output. Outliers are defined as rules that contradict common beliefs. These kind of rules can play an important role in the process of understanding the underlying data as well as in making critical decisions. Outliers Analysis is to uncover the exceptions hidden in the given query output. To perform this over the queried output, we firstly created a preliminary formal context from the given raw data. Then by using `Conexp tool` [13], implication and association rules are generated for complete dataset. These rules shows the correlation among different attributes. After the query is posed, concept lattice of the queried data is created and formal concepts in the form of (extent, intent) tuple are extracted from it. Intents of these formal concepts are then compared with the implication and association rules. If an intent of the queried output is violating any of the implication and association rules, then it is considered as an outlier for that query.

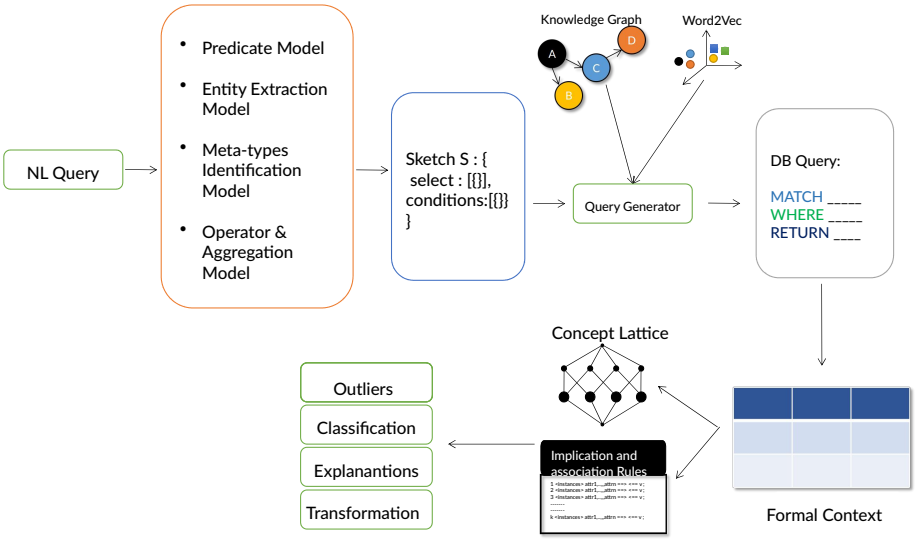


Fig. 1: High Level Architecture of the Process

3.2 Transformation Analysis

This is the second type of analysis that we introduced in our framework. Transformation analysis is used to measure two queries results, where tasks such as conversion of the underlying lattice structure of one set of query results into the lattice structure of another set of query results are required. This kind of analysis is performed by finding the difference between the intents of the formal concepts of both lattices. In our framework when two semantically enriched queries are posed, lattice structures of their respective outputs are generated. To find the possible transformation requirements, we match the intents of both concept lattices and put down the differences between them. This gives us the disparity in the kind of objects contained in both the lattices which will help in transforming one lattice to another.

3.3 Classification analysis

Classification analysis in our framework is done to predict the category of new objects. This is carried out by defining a target attribute **t** in the dataset, generating concept lattices C_i for each value v_i where **i** $\in \mathbb{N}$ of the target attribute and then comparing new object's attributes with the intents of each C_i. In this analysis, a query asking for object details is posed. Lattice structures C_i corresponding to each v_i is stored in the memory. At the run time, matching of new object's attributes set is done with intents of each C_i. If the intent of new object is contained in any one of the lattice C_j for some j $\in$ range(i), then the new

object is classified under the corresponding v_j category otherwise if more than one concept lattices contains the new object's intent then our framework cannot determine its category.

4 Experiments and Results

Census Income dataset taken from UCI machine learning repository [14] is used. This relational database contains 906 observations and 14 features of people like age, occupation, education, salary, workclass, native country etc. We construct the Neo4j knowledge graph from the csv ad also generated the implication and association rules. In this dataset we considered people names as the set of objects and applied conceptual scaling over the multivalued features mentioned above to generate the set of attributes where the objects and the attributes has a binary relation in between them.

Snapshot of the dataset is:

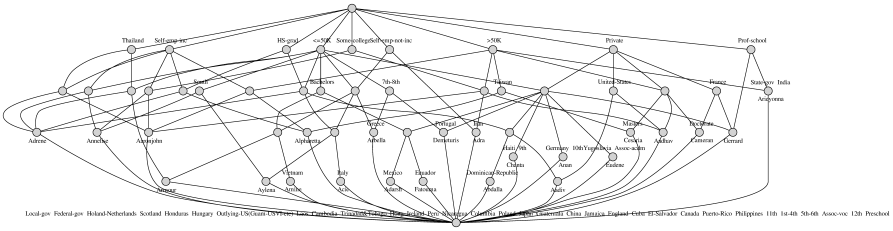
name	age	workclass	education	native country	salary
Aaban	39	State-gov	Bachelors	United-States	<=50K
Aabha	50	Self-emp-not-inc	Bachelors	United-States	<=50K
Aabid	38	Private	HS-grad	United-States	<=50K
Aabriella	53	Private	11th	United-States	<=50K
Aada	28	Private	Bachelors	Cuba	<=50K

Implication and association rules extracted from data are:

S.No.	rule	no. of instances
1	11th $\implies$ $\leq$50K	118
2	State-gov, 5th-6th $\implies$ $\leq$50K	45
3	Private, 10th $\implies$ $\leq$50K	63
4	Doctorate, State-gov $\implies$ >50K	17
5	Federal-gov, Masters $\implies$ >50K	41
6	Local-gov, 12th $\implies$ $\leq$50K	86
7	Bachelors $\implies$ >50K	178

1. Outliers Analysis

Query: List people working more than 60 hours per week and having exceptions in salary with respect to education.



Rules extracted from lattice are:

S.No.	rule
1	Gerrard↔[≤50K,Private,France,Prof-school]⇔Gerrard
2	Arbella↔[>50K,Private,Greece,10th]⇔Arbella⇔Greece
3	Amine↔[≤50K,Self-emp-not-inc,Vietnam,Bachelors]⇔Amine⇔Vietnam
4	Arieyonna↔[>50K,State-gov,India,Prof-school]⇔Arieyonna⇔State-gov,India
5	Adarsh↔[≤50K,Private,Mexico,Bachelors]⇔Adarsh⇔ Mexico
6	Aadhav↔[>50K,Private,United-States,Some-college]⇔Aadhav

Outliers

S.No.	rule
1	Arbella↔[>50K,Private,Greece,10th]⇔Arbella⇔Greece
2	Adarsh↔[≤50K,Private,Mexico,Bachelors]⇔Adarsh⇔ Mexico

Analysis

- Adarsh works >60 hours per week with salary ≤ $ 50 K and Bachelors Degree.
- Arbella works >60 hours per week with salary >$ 50 K and is only 10th grade.

2. Transformation Analysis

Query: What needs to be done to transform workclass, education and salary of men in Cuba to be like men in England?

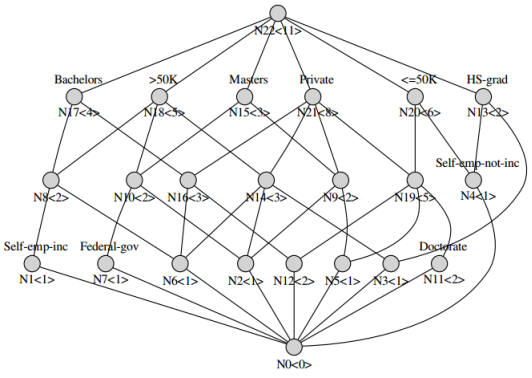


Fig. 2: England

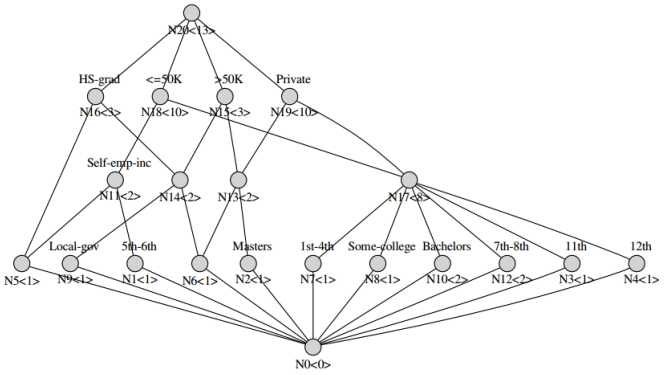


Fig. 3: Cuba

Intents need to be removed are:
a) (≤50K, Self-emp-inc, 5th-6th); b) (Private, >50K, Masters); c) (≤50K, Private, 11th); d) (≤50K, Private, 12th); e)(Private, ≤50K, 7th-8th); and f) (≤50K, Private, 1st-4th)
Intents need to be introduced are:
a) (>50K, Masters, Private); b) (Self-emp-inc, Bachelors, >50K); c) (>50K, Private, HS-grad); d) (Self-emp-not-inc, ≤50K, HS-grad); e) (Private, ≤50K, Masters); f) (Bachelors, >50K, Private); g) (>50K, Masters, Federal-gov); and h) (≤50K, Doctorate, Private)

It shows: Need of higher Education, Need of Self-Employment.

3. Classification Analysis
Query: Predict that whether Aarav has diabetes or not from his blood pressure, body mass index and age.

Person details	**Input from user**
enter name	Aarav
enter age	25
enter Blood Pressure	66
enter Body mass index	23.2

Based on the features of Aarav, it is predicted that he don't have diabetes.

5 Conclusion

We have described a framework wherein the NL sentence is semantically mapped into an intermediate logical form (Sketch) using the framework of multiple sequence tagging networks. This approach of semantic enrichment abstracts the low level semantic information from sentence and helps in generalising into various database queries (e.g. SQL, CQL). Answer of these queries are then further

interpreted using FCA to find out outliers, facts and explanations, classifications and transformations. Experimental results shows that how NLQA and FCA can help an analyst in discovering regularities in a complex data.

References

1. Amit Sangroya, Pratik Saini, Mrinal Rawat, Gautam Shroff, C. Anantaram: Natural Language Business Intelligence Question Answering through SeqtoSeq Transfer Learning, In: DLKT: The 1st Pacific Asia Workshop on Deep Learning for Knowledge Transfer, PAKDD, April(2019)
2. Victor Zhong, Caiming Xiong, Richard Socher: Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning, https://doi.org/arXiv:1709.00103, (2017)
3. Xuezhe Ma, Eduard H. Hovy:End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.,*CoRR,abs/1603.01354*, http://arxiv.org/abs/1603.01354,https://doi.org/1603.01354, *dblp computer science bibliography, https://dblp.org*, (2016)
4. Shefali Bhat, C. Anantaram, Hemant K. Jain: Framework for Text-Based Conversational User-Interface for Business Applications. Knowledge Science, Engineering and Management, In: Second International Conference, KSEM Melbourne, Australia, *DBLP:conf/ksem/2007*, https://doi.org/10.1007/978-3-540-76719-0_31, https://doi.org/10.1007/978-3-540-76719-0_31, *https://dblp.org/rec/bib/conf/ksem/BhatAJ07*, November (2007)
5. Loper, Edward, Bird, Steven: NLTK: The Natural Language Toolkit, In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Volume 1, *ETMTNLP '02*, pages: 63–70, https://doi.org/10.3115/1118108.1118117, https://doi.org/10.3115/1118108.1118117, Philadelphia, Pennsylvania, (2002)
6. Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., McClosky, David: The Stanford CoreNLP Natural Language Processing Toolkit, In: Association for Computational Linguistics (ACL) System Demonstrations,pages: 55–60, http://www.aclweb.org/anthology/P/P14/P14-5010, (2014)
7. Li, Fei, Jagadish, H. V.: Constructing an Interactive Natural Language Interface for Relational Databases, *Proc. VLDB Endow.*, volume: 8, pages: 73–84, http://dx.doi.org/10.14778/2735461.2735468, https://doi.org/10.14778/2735461.2735468, *VLDB Endowment*, September (2014)
8. Lample, Guillaume, Ballesteros, Miguel, Subramanian, Sandeep, Kawakami, Kazuya, Dyera, Chris: Neural Architectures for Named Entity Recognition, In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pages: 260–270, https://doi.org/10.18653/v1/N16-1030 http://aclweb.org/anthology/N16-1030, San Diego, California, (2016)
9. Dmitry I. Ignatov: Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields, Russian Summer School in Information Retrieval, December (2015)
10. K Sumangali, Ch Aswani Kumar: Determination of interesting rules in FCA using information gain, In: First International Conference on Networks and Soft Computing (ICNSC2014), IEEE, August (2014)

11. Peter D. Grnwald: The Minimum Description Length Principle, MIT Press, pages: 3-40, (2007)
12. Bernhard Ganter, Rudolf Wille: Formal Concept Analysis, Mathematical Foundations, Springer, Berlin,Heidelberg,New York, (1999)
13. Serhiy A. Yevtushenko: System of data analysis:Concept Explorer. (In Russian, In: Proceedings of the 7th national conference on Artificial Intelligence KII, pages: 127-134, Russia, (2000)
14. Dua, Dheeru, Graff, Casey: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences, (2017)
15. Ganter B., Wille R.: Formal concept analysis:mathematical foundations. Springer Science & Business Media, (2012)
16. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer: Deep contextualized word representations. CoRR, abs/1802.05365, (2018)
17. Xuezhe Ma, Eduard H. Hovy: Endto-end sequence labeling via bi-directional lstm-cnns-crf, CoRR, abs/1603.01354, (2016)

Using Formal Concept Analysis to Explain Black Box Deep Learning Classification Models

Amit Sangroya, C. Anantaram, Mrinal Rawat, and Mouli Rastogi

TCS Innovation Labs, India
{amit.sangroya, c.anantaram, rawat.mrinal, mouli.r}@tcs.com

Abstract. Recently many machine learning based AI systems have been designed as black boxes. These are the systems that hide the internal logic from the users. Lack of transparency in decision making limits their use in various real world applications. In this paper, we propose a framework that utilizes formal concept analysis to explain AI models. We use classification analysis to study abnormalities in the data which is further used to explain the outcome of machine learning model. The ML method used to demonstrate the ideas is two class classification problem. We validate the proposed framework using a real world machine learning task: diabetes prediction. Our results show that using a formal concept analysis approach can result in better explanations.

1 Introduction

Deep learning techniques have improved the state of the art results in various areas such as natural language processing, computer vision, image processing etc. The area is growing at such a fast pace that everyday a new model is being discovered that improves the state of art rapidly. One of the area that is still under studied is related to the use of these models in real-world such that the outcome can be explained effectively. For instance, if a critical AI (Artificial Intelligence) system such as medical diagnosis only tells whether a patient has a certain disease or not without providing explicit reasons, the users can hardly be convinced of the judgment. Therefore, the ability to explain the decision is an important aspect of any AI system particular natural language processing (NLP) based system.

Recently, lots of works have been done to solve natural language processing research problems such as text classification, sentiment analysis, question answering etc. However, there are very few attempts to explore explainability of such applications. Relational data is usually described by objects and their attributes. Particularly, structure of data is defined by dependencies between the attributes. Explanation consists of performing an exception and transformation analysis to validate the outcome of a ML model. In this paper, our approach to explanation generation is via using formal concept analysis, a conceptually different perspective from existing approaches. A central goal of this research is to build a general purpose or domain-independent framework for interpreting classification outcome of deep learning models, rather than just a single problem in a particular domain. In summary, our contributions in this work are as follows:

– We propose a formal concept analysis based approach in order to generate explanations for the outcomes.
– Furthermore, we show the effectiveness of our method on a real world data set i.e. diabetes prediction.

2 Framework

In this paper, we approach the explanation generation problem from a different perspective – one based on formal concept analysis (FCA). We propose a general concept lattice theory based framework for explanation generation, where given an outcome O of a deep learning model and a domain ontology, the goal is to identify an explanation that can point the user to the prominent feature set f for a certain outcome. We use diabetes classification as an example to evaluate the framework where we model two situations: One where outcome of deep learning black box model and outcome of FCA based classification directly matches and one where it does not match. Further, we present an algorithm, implemented for FCA, that computes such similarities and evaluate its performance experimentally. In addition to providing an alternative approach to solve the explanation generation problem, our approach has the merit of being more generalizable to other problems beyond classification problems as long as they can be modeled using a FCA based concept lattice.

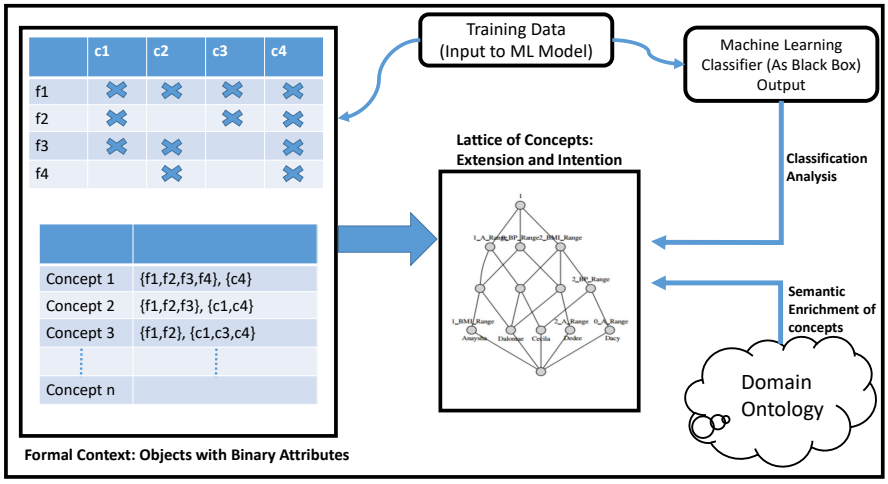


Fig. 1: Overview of the Proposed Framework

Algorithm 1 Explanation of Black Box ML model

Input: M, $c0$, $c1$, $sample\ set\ s$ ▷ M: ML Model; $c0$: lattice of class zero; $c1$: lattice of class one
Output: E ▷ Explanations
procedure PREDICT_FCA($c0$, $c1$, s_i)
    $P \leftarrow \emptyset$ ▷ Prediction
    $class_0\_lattice \leftarrow c0.lattice$
    $class_1\_lattice \leftarrow c1.lattice$
    $s_i_lattice \leftarrow$LOAD_FCA(s)
    for extent, intent $e_j, i_j \in s_i\_lattice$ **do**
        for extent, intent $e_k, i_k \in class_0\_lattice$ **do**
            if $i_k.issubset(i_j)$ **then**
                $P \leftarrow 0$
            end if
        end for
        for extent, intent $e_k, i_k \in class_1\_lattice$ **do**
            if $i_k.issubset(i_j)$ **then**
                $P \leftarrow 1$
            end if
        end for
    end for
    return P
end procedure
procedure EXPLANATIONGENERATOR(S, D)
    $P_{ML} \leftarrow \emptyset$ ▷ ML Predictions
    $P_{FCA} \leftarrow \emptyset$ ▷ FCA Predictions
    $E \leftarrow \emptyset$
    for sample $s_i \in samples$ **do**
        $p \leftarrow M$.predict(s_i)
        if $p > 0.5$ **then**
            P_{ML}.add(1);
        else
            P_{ML}.add(0);
        end if
        P_{FCA}.add(PREDICT_FCA(s_i))
        for feature $f_j \in s_i$ **do**
            $f_j \leftarrow$ MODIFY(f_j)
            $P \leftarrow$ PREDICT_FCA(s_i)
            if $P_{ML_i} == P_{FCA_i}$ **then**
                if $P \neq P_{ML_i}$ **then** E.add (Feature j may be responsible for Sample i classification);
                else E.add (Feature j may not be responsible for Sample i classification);
                end if
            else
                if $P \neq P_{ML_i}$ **then** E.add (Feature j may not be responsible for Sample i classification);
                else E.add (Feature j may be responsible for Sample i classification);
                end if
            end if
        end for
    end for
    return E
end procedure

2.1 Formal Concept Analysis

The fundamental fact underlying FCA is the representability of complete lattices by ordered sets of their meet and join irreducibles. Since ordered sets of irreducibles are naturally represented by binary matrices, this makes it possible to apply certain aspects of the lattice theory to the analysis of data given by object-attribute matrices.

Formal Concept Analysis starts with a formal context (G, M, I) where G denotes an ordered set of objects, M a set of attributes, or items, and I $\subseteq$ G $\times$ M a binary relation between G and M [1]. The statement (g, m) $\in$ I, or gIm, means: "the object g has attribute m". Two operators $(\cdot)'$ define a Galois connection between the power sets (P(G),$\subseteq$) and (P(M),$\subseteq$), with A$\subseteq$G and B$\subseteq$M: $A' = \{m \in M | \forall g \in A : gIm\}$ and $B' = \{g \in G | \forall m \in B : gIm\}$. A pair (A, B), such that A$'$ = B and B$'$ = A, is called a formal concept, where A is called the extent and B the intent of the concept (A, B). The set of all formal concepts of (G, M, I) created by a partial order relation $\leq$, is a subconcept-superconcept hierarchy and is called the concept lattice $\mathcal{L}$.

2.2 Implication Rules

Implication rules S $\Longrightarrow$ T, where S,T $\subseteq$ M holds in context (G,M,I) if S$' \subseteq$ T$'$ i.e., each object having all attributes from S also has all attributes from T. These rules are significant as they expresses the underlying knowledge of interaction among attributes and moreover, also contains statistical values like support and confidence.

2.3 Classification Analysis

Classification analysis is done to predict the category of new as well as existing objects. This is carried out by defining a target attribute in the dataset, generating concept lattices for each value of the target attribute and then comparing new/existing object's attributes with the intents of the concept lattice for each category. In this analysis, a query asking for object details is posed. Lattice structures corresponding to each target value is stored in the memory. Moreover, if an intent *i* of a lattice contains some intent *j* of another lattice, then intent *j* is not considered in the analysis. At the run time, attributes set matching of the new/existing object is done with each of the lattices in the memory. If there is only one lattice *L* whose some concept's intent contains the intent of new/existing object, then the corresponding category is assigned to that object otherwise the result "*not determine*" is declared.

2.4 Semantic Enrichment using Domain Ontology

Ontology is the formal specification of concepts and relationships for a particular domain (e.g. in the domain of finance, US-GAAP is widely used ontology). Ontology has a formal semantic representation that can be used for sharing and reusing knowledge and data. We have downloaded ontology for diabetes from bioportal.bioontology.org/ ontologies/DIAB. In the next step, these concepts and relationships are subsequently coded in the Web Ontology Language (OWL) with Protege.

Table 1: Example of Diabetes Ontology

Subject	**Predicate**	**Object**
type 2 diabetes mellitus	has_exact_synonym	type II diabetes mellitus
type 2 diabetes mellitus	has_exact_synonym	non-insulin-dependent diabetes mellitus
type 2 diabetes mellitus	has_exact_synonym	NIDDM
type 2 diabetes mellitus	is_subClassOf	diabetes mellitus
diastolic blood pressure	has_low_range	< 70
diastolic blood pressure	has_high_range	> 100
body mass index	has_normal_range	< 23

This ontology (stored as a Resource Description Framework graph) stores the concepts of the domain and their relationships with a <subject-predicate-object> structure for each of the concepts. For instance, Table 1 shows an example of diabetes ontology. Here, concepts like diabetes etc. are defined along with concept relationships and synonyms. Additionally, ontology also define the categorical partitioning of diabetic attributes based on medical experts opinion. For example, ontology suggests the normal, low and high ranges for blood pressure. This ontology also assists in deriving implication rules which assists in classification analysis through FCA.

3 Results

The data for diabetes prediction is taken from www.kaggle.com/uciml/pima-indians-diabetes-database. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Data pre-processing involves removing missing/invalid values. Thereafter, we enrich the data using a domain ontology. This involves defining ranges for the records and also building concept hierarchy. Thereafter, we build a ML model to classify if a certain object has diabetes or not. At the same time, we also use FCA approach to classify the same set of objects. Note that the objective of using FCA based classification was just to explain the outcome of ML model, which has been used as a black box. Results are summarized in Table 2.

Table 2: Results using FCA and ML Model

	ML Model	**FCA**
Accuracy	70%	73%
Precision	77%	72%
Recall	63%	90%

3.1 Classification using ML Approach

We used a LSTM based deep neural network based binary classification to train on the processed dataset. Number of training samples were 540 and testing samples were 150. We used all 11 features available in the data such as BMI, Blood Pressure, Insulin etc. The test accuracy of diabetes classification was 70% (See Table 2). Interestingly, accuracy of FCA approach was better. This can be due to the fact that size of dataset was not very large. It might be possible that on a larger dataset ML model might perform better. However, the scope of this work was never to compare the accuracy of two approaches, but to use FCA based lattice theory to explain the output of black box ML model. The explanation of the outcomes was generated using FCA model as explained in the next subsection.

3.2 Classification using FCA Approach

We divided the training data (the same data that was used in ML model) into two classes: diabetes and no diabetes. Then, we created two separate concept lattices for both classes as shown in Figure 2 and 3. For each sample in test set, we created its lattice alongwith extent and intent of each concept in the lattice. Thereafter, we compared the intents of concept in sample lattice with concept in both lattices (class lattices i.e. lattices of diabetes and no diabetes). The comparison is based on subset matching between sample lattice and class lattices. Wherever there is a match between lattices, that class is assigned as predicted class for the test sample.

3.3 Explaining the ML Model Outcome using FCA

We compared the outcomes of ML model and FCA based classification. We take each sample in the test set and we try to map to the feature set. The goal of explanation is to identify the feature which may be prominent to classify a given sample into a particular class. In order to achieve this, we change the features and observe the outcome with modified features. If the outcome with modified features change (i.e. changing a feature fi, leads to change in Outcome Oj), we can assert that fi is responsible for the outcome (See Algorithm 1 for details).

Table 3 shows the identified feature set for two classes. It shows the relative importance of each feature for identifying a sample into diabetes or no diabetes. In the scope of current work, we present the results with individual features only. Similar experiments can be performed to compute the feature sets as well. As we observe, *Age* is least important feature for an outcome of diabetes class, whereas *Blood Pressure* is most important feature. Similarly, for an outcome to be in non-diabetes class *BMI* is the most prominent feature.

Outcome (Based on the features and Implication rules): Aarav doesn't have diabetes.

In order to qualitatively evaluate the results, we identified implication rules from the training data as shown in Table 5. For a given test sample, we also used implication rules to validate the output. For Example: *Predict that whether Aarav has diabetes or not from his blood pressure, body mass index and age (See Table 4*).

Table 3: Feature Interpretation for two classes (Diabetes and Non Diabetes)

	Diabetes	**Non Diabetes**
Number of times of pregnancy — (# Preg)	15.6%	36.7%
Plasma glucose concentration every 2 hours in an oral glucose tolerance test — (Plasma)	13.2%	37.5%
diastolic blood pressure (mm Hg) —- (Diast BP)	16.4%	42.18%
triceps skin fold thickness (mm) —- (skin)	12.5%	41.4%
2-Hour serum insulin (mu U/ml) —- (insulin)	11.7%	43.7%
body mass index (weight in kg/(height in (mm)2) —- BMI	10.9%	45.3%
diabetes pedigree function — Pedigree	9.3%	43.7%
Age in years —- Age	5.4%	40.6%

Table 4: Classification Example using FCA

Person details	**Input from user**
Name	Aarav
Age	25, Age-range(2)
Blood Pressure	66, BP-range(1)
Body Mass Index	23.2, BMI-Range(2)

4 Related Work

Most machine learning model rely on validation set accuracy as a way of primary measurement of trust. However, there are limitations of these approaches in using models in a real world paradigm. Recognizing the importance of interpretations in assessing trust, various frameworks have been proposed that focus on interpretable models, especially for the medical domain [2,3,4]. While such models may be appropriate for some domains, they may not apply equally well to others. In the domain of computer vision, systems that rely on object detection to produce candidate alignments [5] or attention [6] are able to produce explanations for their predictions. However these models are constrained to specific neural network architectures. Our focus is on building general, model-agnostic explanations that can be applied to any classifier.

Another common approach for generating explanation is to build another model over the outcome of original model [7,8]. One limitation of this approach is that these models approximate the original model globally, thus interpreting outcomes at a fine grain level becomes a significant challenge. In order to interpret model at fine grain local level, LIME is a promising approach that approximates the original model locally [9]. Recent works such as SHAP (SHapley Additive exPlanations) provide robust framework to interprete predictions of Ml models [10]. Machine learning models have also been described in terms of Formal Concept Analysis (FCA) [11]. Similarly, Formal concept analysis has been successfully used in other areas such as knowledge processing [12].

Table 5: Implication rules

Rule	**# instances**
BP-range(2), Age-range(2) $\Longrightarrow$ Outcome(0)	226
BMI-range(1), BP-range(1) $\Longrightarrow$ Outcome(0)	128
BMI-Range(2), BP-Range(2) $\Longrightarrow$ Outcome(1)	63
Age-Range(1), BMI-Range(2), BP-Range(1) $\Longrightarrow$ Outcome(1)	41
BP-Range(0), Age-Range(2), BMI-Range(0) $\Longrightarrow$ Outcome(0)	95
BP-Range(0), Age-Range(2), BMI-Range(2) $\Longrightarrow$ Outcome(1)	86
BP-Range(1), Age-Range(1), BMI-Range(2) $\Longrightarrow$ Outcome(1)	178

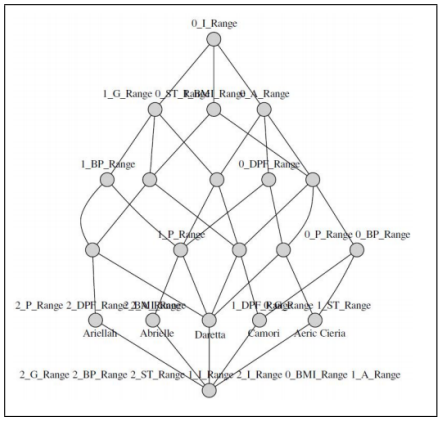


Fig. 2: No Diabetes

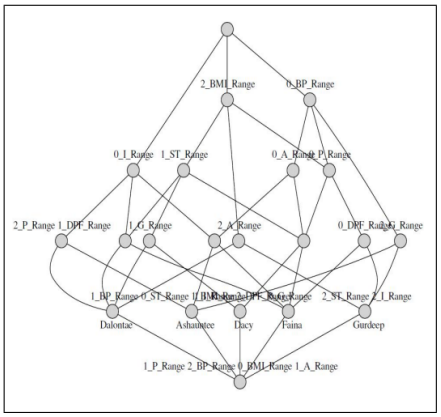


Fig. 3: Diabetes

Our approach is model and domain agnostic. However, using FCA based interpretation approach, the outcome can be interpreted with a sound theoretical basis.

5 Conclusion and Future Work

We considered Formal Concept Analysis in context of interpretation of machine learning models particularly focusing on classification and assuming that model to be explained is a black box model. The main attention was drawn to the lattice based classification analysis of attributes. We showed the significance using well known classification problem i.e. diabetes prediction. In this paper, we limited our experiments to two class classification problems, however the proposed approach can be generalized to multi-class classification problems easily. In future, we want to extend this work to various other domains such as computer vision.

References

1. B. Ganter and R. Wille, *Formal Concept Analysis, Mathematical Foundations*. Berlin,Heidelberg,New York: Springer, 1999.

2. R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1721–1730. [Online]. Available: http://doi.acm.org/10.1145/2783258.2788613
3. B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "An interpretable stroke prediction model using rules and bayesian analysis," in *Proceedings of the 17th AAAI Conference on Late-Breaking Developments in the Field of Artificial Intelligence*, ser. AAAIWS'13-17. AAAI Press, 2013, pp. 65–67. [Online]. Available: http://dl.acm.org/citation.cfm?id=2908286.2908308
4. B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Mach. Learn.*, vol. 102, no. 3, pp. 349–391, Mar. 2016. [Online]. Available: https://doi.org/10.1007/s10994-015-5528-6
5. A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 664–676, Apr. 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2598339
6. K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 2048–2057. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045336
7. D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, Aug. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1859912
8. M. W. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Proceedings of the 8th International Conference on Neural Information Processing Systems*, ser. NIPS'95. Cambridge, MA, USA: MIT Press, 1995, pp. 24–30. [Online]. Available: http://dl.acm.org/citation.cfm?id=2998828.2998832
9. M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778
10. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. USA: Curran Associates Inc., 2017, pp. 4768–4777. [Online]. Available: http://dl.acm.org/citation.cfm?id=3295222.3295230
11. S. O. Kuznetsov, "Machine learning on the basis of formal concept analysis," *Automation and Remote Control*, vol. 62, no. 10, pp. 1543–1564, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1012435612567
12. J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene, "Review: Formal concept analysis in knowledge processing: A survey on applications," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6538–6560, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.05.009

Validating Correctness of Textual Explanation with Complete Discourse trees

Boris Galitsky
Oracle Corp Redwood Shores CA USA
And
Dmitry Ilvovsky
National Research University Higher School of Economics, Moscow, Russia

Abstract

We explore how to validate the soundness of textual explanations in a domain-independent manner. We further assess how people perceive explanations of their opponents and what are the factors determining whether explanations are acceptable or not. We discover that what we call a *complete discourse tree* (complete DT) determines the acceptability of explanation. A complete DT is a sum of a traditional DT for a paragraph of actual text and an imaginary DT for a text about entities used but not explicitly defined in the actual text.

1 Introduction

Providing explanations of decisions for human users, and understanding how human agents explain their decisions, are important features of intelligent decision making and decision support systems. A number of complex forms of human behavior is associated with attempts to provide acceptable and convincing explanations. In this paper, we propose a computational framework for assessing soundness of explanations and explore how such soundness is correlated with discourse-level analysis.

Importance of the explanation-aware computing has been demonstrated in multiple studies and systems. Also, (Walton, 2007) argued that the older model of explanations as a chain of inferences with a pragmatic and communicative model that structures an explanation as a dialog exchange. The field of explanation-aware computing is now actively contributing to such areas as legal reasoning, natural language processing and also multi-agent systems (Dunne and Bench-Capon, 2006). It has been shown (Walton, 2008) how the argumentation methodology implements the concept of explanation by transforming an example of an explanation into a formal dialog structure. Galitsky (2008) differentiated between explaining as a chain of inference of facts mentioned in dialogue, and meta-explaining as dealing with formal dialog structure represented as a graph. Both levels of explanations are implemented as argumentation: explanation operates with individual claims communicated in a dialogue, and meta-explanation relies on the overall argumentation structure of scenarios.

In this paper we explore how good explanation in text can be computationally differentiated from bad explanation. Intuitively, a *good* explanation convinces the addressee that a communicated claim is right, and it involves valid argumentation patterns, logical, complete and thorough. A bad explanation is unconvincing, detached from the beliefs of the addressee, includes flawed argumentation patterns and omits necessary entities. In this work we differentiate between good and bad explanation based on a *human response* to such explanation. Whereas users are satisfied with good explanation by a system or a human, bad explanations usually lead to dissatisfactions, embarrassment and complaints.

2 Validating explanations with Discourse Trees

2.1 Classes of explanation

To systematically treat the classes of explanation, we select an environment where customers receive explanations from customer service regarding certain dissatisfactions these customers encountered. If these customers are not satisfied with explanations, they frequently submit detailed complaints to consumer advocacy sites. In some of these complaints these customers explain why they are right and why the company's explanation is wrong. From these training sets we select the *good/bad* explanation pairs and define respective explanation classes via learning to recognize them.

Another way to consider a bad explanation is what we call an *explanation attempt*: a logical chain is built but it has some omissions and inconsistencies so that the explanation is bad. An absense of a logical chain means the absense of explanation; otherwise, if such chain obeys certain logical properties it can be interpreted by something else besides explanation but instead argumentation, clarification, confirmation or other mental or epistemic state.

2.2 Explanation and Argumentation

Explanations are correlated with argumentation and sentiments. A request to explain is usually associated with certain arguments and a negative sentiment.
For an arbitrary statement S a person may have little or no prior reason for believing this statement to be true. In this case a cognitive response is a doubt, which is articulated with a request for evidence.
Evidence is a kind of reason, and the attempt to provide evidence in support of a conclusion is normally called an *argument*. Argument reasoning is represented on the top of Fig. 1.

On the other hand a person may already know S and require no further evidence for the truth of S. But she still may not understand why S holds (occurred, happened etc. In this case she would request for a cause. Explanation is defined as an attempt to provide a cause in support of a conclusion. Explanation reasoning may be represented in the bottom of Fig. 1.

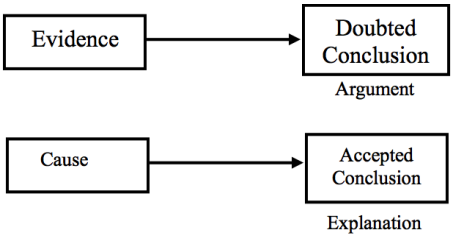


Fig.1: Relationship between argumentation and explanation

2.3 Hybrid discourse trees

In the banking domain *nonsufficient fund fee* (NSF) is a major problem that banks have difficulties communicating with customers. An example of brief, informal explanation follows:

It's not always easy to understand overdraft fees. When a transaction drops your checking account balance below zero, what happens next is up to your bank. A bank or credit union might pay for the transaction or decline it and, either way, could charge you a fee.

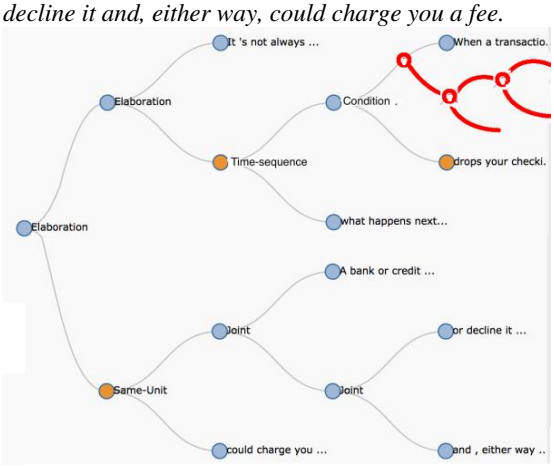


Fig 2. Discourse tree of explanation text with the imaginary part shown in the top-right for nucleus '*transaction*'.

The concept of *transaction* is not tackled in this text explaining nonsufficient fee. An ontology could specify that transaction = *{wiring, purchasing, sending money}* but it is hard to be complete. Instead, one can complement the notion of transaction via additional text that will elaborate on transaction, providing more details on it.

Hence *Elaboration* relation for nucleus *transaction* is not in actual DT but is assumed by a recipient of this explanation text. We refer to such rhetorical relations as Imaginary: they are not produced from text but are instead induced by the context of explanation. Such multiple imaginary RRs form additional nodes of an actual DT for a text being communicated. We refer to the extended DT as *complete*: it combines the actual DT and its imaginary parts. Naturally, the latter can be dependent on the recipient: different people keep in mind distinct instances of *transactions*.

We formalize this intuition by using discourse structure of the text expressed by DTs. Arcs of this tree correspond to rhetorical relations (RR), connecting text blocks called Elementary Discourse Units (EDU). We rely on the Rhetorical Structure Theory (RST, Mann and Thompson, 1988) when construct and describe discourse structure of the text.

When people explain stuff, they do not have to enumerate all premises: some of them implicitly occurring in the explanation chain and are assumed by the person providing explanation to be known or believed by an addressee. However, a DT for a text containing explanation only incudes EDUs from actual text and assumed, implicit parts with its entities and phrases (which are supposed to enter explanation sequence) are absent. How can we cover these implicit entities and phrases?

In the considered example *Elaboration* relation for nucleus *transaction* is not in actual CDT but is assumed by a recipient of this explanation text. We refer to such rhetorical relations as *Imaginary*: they are not produced from text but are instead induced by the context of explanation. Such multiple imaginary RRs form additional nodes of an actual DT for a text being communicated. We refer to the combined CDTs as *hybrid*: it combines the actual CDT and its imaginary parts. Naturally, the latter can be dependent on the recipient: different people keep in mind distinct instances of *transactions*. Complete discourse tree for the example is shown on Fig.2. Complete discourse trees also have communicative actions attached to their edges in the form of VerbNet verb signatures (Galitsky and Parnis, 2019).

2.4 Semantic representation

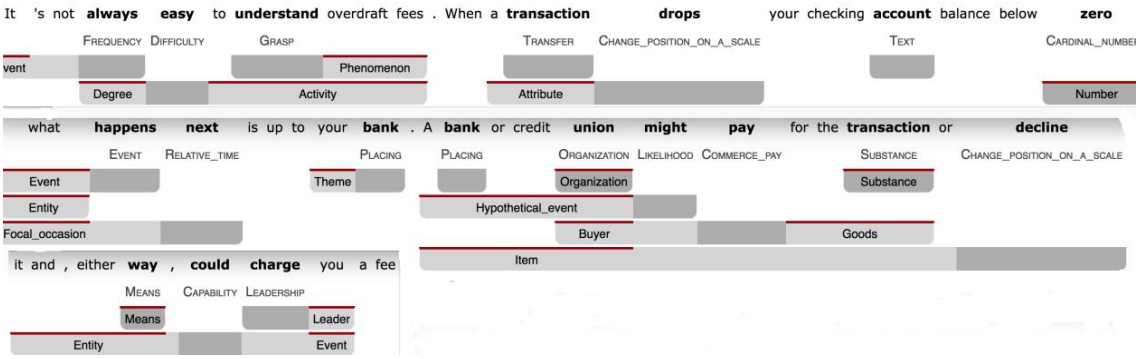


Fig.3 Frame semantic parse for the explanation

A frame semantic parse for the same text is shown in Fig. 3. The reader observes that it is hard to tag entities and determine context properly. *Bank* is tagged as Placing (not disambiguated properly) and '*credit union might*' is determined as a hypothetical event since *union* is represented literally, as an *organization*, separately from *credit*. Overall, the main expression being explained, '*transaction drops your checking account balance below zero',* is not represented as a cause of a problem by semantic analysis, since a higher level considerations involving a banking – related ontology would be required.

Instead of relying on semantic–level analysis to classify explanations, we propose a discourse-level machinery. This machinery allows including the explanation structure beyond the ones from explanation text but also from the accompanying texts mined from various sources to obtain a complete logical structure of the entities involved in explanation.

2.5 Discourse tree of explanations

Valid explanation in text follow certain rhetoric patterns. In addition to default relations of Elaborations, valid explanation relies on *Cause, Condition*, and domain-specific *Comparison* (Fig. 4) As an example, we provide an explanation for why *thunder sound comes after lightning*:

'We see the lightning before we hear the thunder. This is because light travels faster than sound. The light from the lightning comes to our eyes much quicker than the sound from the lightning. So we hear it later than we see it.'

```
joint
  elaboration (LeftToRight)
    cause (LeftToRight)
      temporal (LeftToRight)
        TEXT: We see the lightning
        TEXT: before we hear the thunder .
      TEXT: This is because light travels faster than sound .
    elaboration (LeftToRight)
      TEXT: The light from the lightning travels to our eyes much quicker than the sound from the
lightning .
      comparison (LeftToRight)
        TEXT:so we hear it later
        TEXT:than we see it .
```

Fig. 4: A discourse tree for an explanation of a lightning

The clause we need to obtain for an implication in the explanation chain is verb-group-for-moving {*moves, travels, comes*} *faster* → *verb-group-for-moving-result* {*earlier*}. This clause can be easily obtained by web mining, searching for expression 'if noun verb-group-for-moving *faster* then noun *verb-group-for-moving-result earlier.*

What would make this DT look like a one for invalid explanation? If any RR under top-level *Elaboration* turns into *Joint* it would mean that the explanation chain is interrupted.

We explore argumentation structure example of (Toulmin, 1958, Kennedy et al., 2006). We show two visualizations of the discourse tree and the explanation chain (in the middle) in Fig. 5.

```
  elaboration
    TEXT:Harry was born in Bermuda .
    explanation (LeftToRight)
      attribution (LeftToRight)
        TEXT:A person born in Bermuda is a British subject .
        TEXT:It is on account of the following statutes 123 .
      condition
        TEXT:So , presumably , Harry is a British subject ,
        Joint
          TEXT:unless both his parents were aliens ,

          TEXT:or he has become a naturalized American .
```

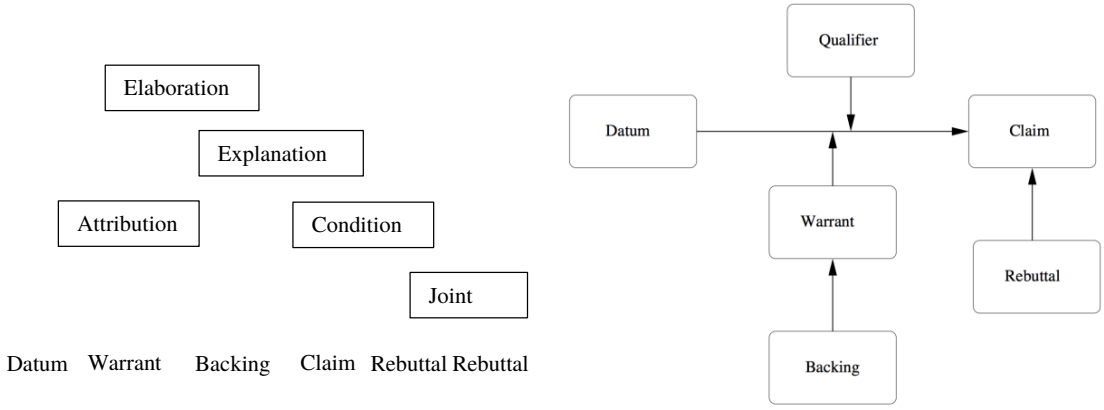


Fig. 5: Toulmin's argument structure (in the middle) and its rhetorical representation via EDUs (on the top) and via discourse relations (on the bottom)

An interesting application of Toulmin's model is the argumentative grammar by Lo Cascio (1991), a work that, by defining associative rules for argumentative acts, is naturally applicable, and indeed has been applied, to the analysis of discourse structure in the pre-DT times.

2.6 Logical Validation of Explanation via Discourse trees

Logically, explanation of text *S* is a chain of premises $P_1, \ldots, P_m$ which imply S. S is frequently referred to as a subject of explanation. For this chain $P_1, \ldots, P_m$ each element P_i is implied by its predecessors: $P_1, \ldots P_{i-1} \Rightarrow P_i$. In terms of a discourse tree, there should be a path in it where these implications are realized via rhetorical relations. We intend to define a mapping between EDUs of a DT and entities P_i occurring in these EDUs which form the explanation chain. In terms on underlying text, P_i are entities or phrases which can be represented as logical atoms or terms.

These implication-focused rhetorical relations *rr* are:

1) *elaboration*: P_i can be an elaboration of P_{i-1} ;
2) *attribution*: P_i can be attributed to P_{i-1} ;
3) *cause*: this is a most straightforward case,

Hence $P_i \Rightarrow P_j$ if $rr(EDU_i, EDU_j)$ where $P_i \in EDU_i$ and $P_j \in EDU_j$. We refer to this condition as "*explainability" via Discourse Tree*.

Actual sequence $P_1, \ldots, P_m$ for S is not known, but for each S we have a set of good explanations $P_{g1}, \ldots, P_{gm}$ and a set of bad explanations $P_{b1}, \ldots, P_{b2}$.

Good explanation sequences obey *explainability via DT* condition and bad – do not (Galitsky 2018). Bad explanation sequences might obey *explainability via DT* condition for some P_{bi}. If a DT for a text is such that *explainability via DT* condition does not hold for any P_{bi} then this DT does not include any explanation at all.

The reader can observe that to define a good and a bad explanation via a DT one needs a training set covering all involved entities and phrasing Pi occurring in both positive and negative training sets.

2.7 Constructing Imaginary Part of a Discourse Tree

By our definition imaginary DTs are the ones not obtained from actual text but instead built on demand to augment the actual ones. For a given chain $P_1, \ldots, P_i', \ldots, P_m$ let P_i' be the entity which is not explicitly mention in a text but instead is assumed to be known to the addressee. This P_i' should occur in other texts in a training dataset. To make the *explainability via DT* condition applicable, we need to augment actual DT_{actual} with imaginary $DT_{imaginary}$ such that $P_i' \in$ EDU of this $DT_{imaginary}$. We denote DT_{actual} $DT_{imaginary}$ as $DT_{complete}$.

If we have two textual explanations in the positive set of good explanations for the same S, T_1 and T_2:

T_1: P_1 ,..., $P_m \Rightarrow$ S
T_2: P_1 , P_i', ..., $P_m \Rightarrow$ S

then we can assume that P_i' should occur in a complete explanation for S and since it does not occur in T_1 then DT(T_1) should be augmented with $DT_{imaginary}$ such that $P_i' \in$ EDU of this $DT_{imaginary}$.

3 Learning Framework and Evaluation

In this section we automate our validation of text convincingness including description of a training dataset and learning framework.

We conduct our evaluation in two steps. Firstly, we try to distinguish between texts with explanation and without explanation. This task can be accomplished without an involvement of virtual DTs. Secondly, once we confirm that that can be done reasonably well, we drill into more specific tasks of differentiating between good and bad explanation chains within the dataset of the first task.

3.1 Building a Dataset of Good/bad Explanation Chains

We form the positive explanation dataset from the following sources:

1. Customer complaints;
2. Paragraphs from physics and biology textbook;
3. Yahoo! Answers for *Why/How-to* questions.

The negative training dataset includes the sources of a totally different nature:

1. Definition/factoid paragraphs from Wikipedia, usually, first paragraphs;
2. First paragraphs of news articles introducing new events;
3. Political news from Functional Text Dimension dataset.

We formed the balances components of the positive and negative dataset for both tasks: each component includes 240 short texts 5-8 sentences (250-400 words).

We now comment on each source. The purpose of the customer complaint dataset is to collect texts where authors do their best to explain their points across by employing all means to show that they are right and their opponents are wrong. Complaints are emotionally charged texts providing explanation of problems they encountered with a financial service, how they tried to explain their viewpoint to a company and also a description of how these customers attempted to solve it (Galitsky et al., 2008, GitHub Customer Complaints dataset 2019).

Also, to select types of text with and without explanation, we adopt the genre system and the corpora from (Lee, 2001). The genre system is constructed relying on the Functional Text Dimensions. These are genre annotations which reflect judgments as to what extent a text can be interpreted as belonging to a generalized functional category. A genre is a combination of several dimensions. For the positive dataset, we select the genre with the highest density of explanation such as scientific textbook. For the negative dataset, we focus on the genres which are least likely to contain explanations, such as advertisement, fiction-prose, instruction manuals and political news. The last one is chosen since it has the least likelihood to contain an explanation.

For the positive dataset for the second task, as good explanation chains, we rely on the following sources:

1. Customer complaints with *valid* argumentation patterns;
2. Paragraphs from phisics textbook explaining certain phenomena, which are neither factoid nor definitional;
3. Yahoo! Answers for *Why/How-to* questions;

We form the negative dataset from the following sources:

1. Customer complaints with *invalid* argumentation patterns; these complaints are inconsistent, illogical and rely on emotions to bring their points across;
2. Paragraphs from phisics textbook formulating longer questions and problems;
3. Yahoo! Answers for *Why* (not *How-to*) questions which are reduced to break the explanation flow. Sentences are deleted or re-shuffled to produce an incohesive, non-systematic explanation.

3.2 Crawling Information for Imaginary Discourse Tree Construction

Imaginary DTs can be found by employing background knowledge in a domain independent manner: no offline ontology construction is required. Documents that were found on the web can be the basis of constructing imaginary DTs following the algorithm described in the Section 2.4.

Given an actual part of the text A, we outline a top-level search strategy for finding a source for imaginary DTs (background knowledge) B.

1) Build DT for A;
2) Obtain pairs of entities from A that are not linked in DT (e.g. *thunder, eye*);
3) Obtain a set of search queries based on provided pairs of entities
4) For each query:
   a) Find a short list of candidate text fragments on the web using search engine API (such as Bing);
   b) Build DT for the text fragments;
   c) Select fragments which contain rhetoric relation (*Elaboration, Attribution, Cause*) linking this pair of entities;
   d) Choose the fragment with the highest relevance score

The *entity* mentioned in the algorithm can be interpreted in a few possible ways. It can be named entity, head of a noun phrase or a keyword extracted from a dataset.

Relevance score can be based on the score provided by the search engine. Another option – computing score based on structural discourse and syntactic similarity (Galitsky, 2017).

3.3 Learning Approaches and Pipelines

Discourse Tree Construction. A number of RST parsers constructing discourse tree of the text are available at the moments. For instance, in our previous studies we used the tool provided by (Surdeanu et.al., 2015) and (Joty et al., 2014).

Nearest Neighbor learning. To predict the label of the text, once the complete DT is built, one needs to compute its similarity with DTs for the positive class and verify that it is lower than similarity to the set of DTs for its negative class. Similarity between CDT's is defined by means of maximal common sub-DTs. Formal definitions of labeled graphs and domination relation on them used for construction of this operation can be found, e.g., in (Ganter, 2001).

SVM Tree Kernel learning. A DT can be represented by a vector of integer counts of each sub-tree type (without taking into account its ancestors). For Elementary Discourse Units (EDUs) as labels for terminal nodes only the phrase structure is retained: we suppose to label the terminal nodes with the sequence of phrase types instead of parse tree fragments. For the evaluation purpose Tree Kernel builder tool (Moschitti, 2006) can be used.

3.4 Detecting explanations and valid explanation chains

We first focus on the first task, detecting paragraphs of text which contain explanation, and estimate the detection rate in Table 1. We apply two different learning techniques, nearest neighbor (in the middle, greyed) and SVM TK, applied to the same discourse-level and syntactic data.

Table 1: Explanation detection rate

Source	P_{KNN}	R_{KNN}	$F1_{KNN}$	P_{SVM}	R_{SVM}	$F1_{SVM}$
1^+ vs 1^-	77.3	80.8	79.0	80.9	82.0	81.4
2^+ vs 2^-	78.6	76.4	77.5	74.6	74.8	74.7
3^+ vs 3^-	75.0	77.6	76.3	76.6	77.1	76.8
$1..3^+$ vs $1..3^-$	76.8	78.9	77.8	74.9	75.4	75.1

The highest recognition accuracy, reaching 80%, is achieved for the first pair of the dataset components, complaints vs wikipedia factois, most distinct 'intense' explanation vs enumeration of facts, with least explanations. The other datasets deliver 2-3% drop in recognition performance. These accuracies are comparable with various tasks in genre classification (one-against-all setting in Galitsky et al., 2016).

Table 2 shows the results of differentiation between good and bad explanation. The accuracy is about 12% lower than for the first task, since the difference between the good and bad explanation in text is fairly subtle.

Table 2: Recognizing good and bad explanation chains

Source	$P_{-virtual}$	$R_{-virtual}$	$F1_{-virtual}$	P	R	F1
1^+ vs 1^-	64.3	60.8	62.5	72.9	74.0	73.4
2^+ vs 2^-	68.2	65.9	67.0	74.6	74.8	74.7
3^+ vs 3^-	63.7	67.4	65.5	76.6	77.1	76.8
$1..3^+$ vs $1..3^-$	66.4	64.6	65.5	74.9	75.4	75.1

However, validation of explanation chain is an important task in a decision support. A low accuracy can still be leveraged by processing a large number of documents and detecting a birst in problematic explanation in a corpus of texts.

4 Discussion and Conclusions

In this work we considered a new approach to validating the convincingness of textual explanations. We introduced the notion of a *complete discourse tree* (complete DT) including actual and imaginary parts. Imaginary DT is constructed for the text about entities used but not explicitly defined in the actual text.

We outlined an algorithm for building an imaginary discourse tree. We also described a possible strategy for crawling background knowledge which is the source of the imaginary part. We also introduced the new dataset of good and bad explanations made by complainants in the financial domain. Finally, we outlined the learning framework used for automated detection of good and bad explanations. It is based on RST parsing and learning on complete discourse trees provided by the parser.

Both professional and non-professional writers provide explanations in texts but detection of invalid explanations is significantly harder in the former case compared to the latter. Professional writers in such domains as politics and business are capable of explaining “anything”, and in user-generated content errors are visible.

Detecting faulty explanations in user-generated content is important in automated Customer Relation Management systems where a response to user requests with valid explanation should be different to user response with invalid explanation.

It is important to combine rule-based learning frameworks with the ones with implicit feature engineering such as statistical and deep learning. The latest history of applications of statistical technique sheds a light on the limitation of these techniques for systematic exploration of a given domain. Once statistical learning delivered satisfactory results for discourse parsing, the interest to automated discourse analysis faded away. Since the researches in statistical ML for discourse parsing were mainly interested in recognition accuracies and not the interpretability of obtained DTs, no further attempts at leveraging obtained DTs were made. However, a number of studies including the given one demonstrate that DTs provide insights in the domain where keyword statistics does not help.

On the basis of work by Austin, Searle, Grice and Lorenzen, such discipline as pragmadialectics provides a comprehensive analysis of argumentative dialogues. This discipline combines the study on the formalism to represent data, from modern logic, and empirical observations, from descriptive linguistics, for the analysis of argumentative dialogues, modeled by dialectics, seen as sets of linguistic speech. The model proposes rule-base argumentative dialogues, but does not help with a dialogue generation algorithm.

Acknowledgements

The work of Dmitry Ilvovsky was supported by the Russian Science Foundation under grant 17-11-01294'.

References

Mann, William and Sandra Thompson. 1988. *Rhetorical structure theory: Towards a functional theory of text organization. Text* - Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.

Explanation on Wikipedia. 2018. https://en.wikipedia.org/wiki/Explanation#Meta-explanation.

Galitsky, B., Kuznetsov SO. 2008. Learning communicative actions of conflicting human agents. J. Exp. Theor. Artif. Intell. 20(4): 277-317.

Galitsky B (2018) Customers' Retention Requires an Explainability Feature in Machine Learning Systems They Use. AAAI Spring Symposium Series.

Jansen, P., M. Surdeanu, and P. Clark. 2014. Discourse Complements Lexical Semantics for Nonfactoid Answer Reranking. ACL.

Surdeanu, M., Thomas Hicks, and Marco A. Valenzuela-Escarcega. 2015. Two Practical Rhetorical Structure Theory Parsers. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies: Software Demonstrations (NAACL HLT), 2015.

Galitsky, B, D Ilvovsky, SO Kuznetsov (2016) Style and Genre Classification by Means of Deep Textual Parsing. Computational Linguistics and Intellectual Technologies: DIALOG, Moscow, Russia.

Galitsky, B. 2017. Matching parse thickets for open domain question answering, Data & Knowledge Engineering, Volume 107, January 2017, Pages 24-50.

Galitsky B, D Ilvovsky, SO Kuznetsov (2018) Detecting logical argumentation in text via communicative discourse tree. Journal of Experimental & Theoretical Artificial Intelligence 30 (5), 637-663.

Galitsky B, Parnis A (2018) Accessing Validity of Argumentation of Agents of the Internet of Everything. Artificial Intelligence for the Internet of Everything, 187-216.

Joty, S., Moschitti, A. 2014 Discriminative reranking of discourse parses using tree kernels. EMNLP 2014.

Ganter, B., Kuznetsov, S.O. 2001. Pattern structures and their projections. In: International Conference on Conceptual Structures. pp. 129-142. Springer.

Moschitti, A. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany.

Paul E. Dunne and Trevor J. M. Bench-Capon. 2006. Computational Models of Argument: Proceedings of COMMA 2006, IOS Press, 2006.

Lo Cascio, V. 1991. Grammatica dell'Argomentare: strategie e strutture [A grammar of Arguing: strategies and structures]. Firenze: La Nuova Italia.

Walton, D. 2007. Dialogical Models of Explanation. Explanation-Aware Computing: Papers from the 2007 AAAI Workshop, Association for the Advancement of Artificial Intelligence, Technical Report WS-07-06, AAAI Press, 2007,1-9.

Walton, D., Reed, C., Macagno, F. 2008. Argumentation schemes. Cambridge University Press

Lee, David YW. Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the BNC jungle. (2001)

Kennedy, X.J., Dorothy M. Kennedy, and Jane E. Aaron.. "Reasoning". The Bedford Reader. 9th ed. New York: Bedford/St. Martin's, 2006. p. 519–522.

Toulmin, S. The Uses of Argument. Cambridge At the University Press, 1958.

Least General Generalization of the Linguistic Structures

Boris Galitsky
Oracle Corp., Redwood Shores CA USA
and
Dmitry Ilvovsky
National Research University Higher School of Economics, Moscow, Russia

Abstract

We convert existing training datasets into the ones closed under linguistic generalization operations to expand infrequent cases. We transfer the definition of the least general generalization from logical formulas to linguistic structures, from words to phrases, sentences, speech acts and discourse trees. The main advantage of the resultant frameworks is explainability and learnability from a small set of samples. Learning via generalization of linguistic structures turned out to be well suited for industrial linguistic applications with limited training datasets.

1 Introduction

A lack of data, especially covering tail phenomena, is a major bottleneck for language learning system. As statistical and deep learning language systems provide higher overall accuracy in most cases, it is never obvious how to circumscribe these successful cases and how to extend the training datasets to cover tail, unsuccessful cases (Ettinger et al., 2017., Kovalerchuk and Kovalerchuk, 2017). To address this problem we can expand a training dataset into a form that would force the learning framework to acquire generalizations from it.

To measure of similarity of abstract entities expressed by logic formulas, a least-general generalization was proposed for a number of machine learning approaches, including explanation based learning and inductive logic programming. Least general generalization was originally introduced by (Plotkin, 1970). It is the opposite of most general unification (Robinson, 1965) therefore it is also called anti-unification.

Least general generalization can be considered as an intersection operation in the FCA framework (Ganter and Kuznetsov, 2001). This operation can be used for the construction of the lattices of the closed sets in many domains such as text analysis (Makhalova et al., 2015) and many others (Kuznetsov, 2013a,2013b).

2 Generalization of the Texts

For instance, for the two words of the same POS, their generalization is the same word with POS. If lemmas are different but POS is the same, POS stays in the result. If lemmas are the same but POS is different, lemma stays in the result.

Let us represent a meaning of two natural language expressions by logic formulas and then construct unification and anti-unification of these formulas. Some words (entities) are mapped into predicates, some are mapped into their arguments, and some other words do not explicitly occur in logic form representation but indicate the above instantiation of predicates with arguments. How to generalize the expressions?

- camera with digital zoom
- camera with zoom for beginners

To express the meanings we use logic predicates *camera(name_of_feature, type_of_users)* (in real life, we would have much higher number of arguments), and *zoom(type_of_zoom)*. The above NL expressions will be represented as:

camera(zoom(digital), AnyUser)

camera(zoom(AnyZoom), beginner),

where variables (non-instantiated values, not specified in NL expressions) are capitalized. Given the above pair of formulas, unification computes their most general specialization *camera(zoom(digital), beginner),* and anti-unification computes their most specific generalization, *camera(zoom(AnyZoom), AnyUser)*.

At syntactic level, we have generalization ('^') of two noun phrases as: {NN-camera, PRP-with, [digital], NN-zoom [for beginners]}.

We eliminate expressions in square brackets since they occur in one expression and do not occur in another. As a result, we obtain{NN-camera, PRP-with, NN-zoom]}, which is a syntactic analog as the semantic generalization above.

The purpose of an abstract generalization is to find commonality between portions of text at various semantic levels. Generalization operation occurs on the levels of Text / Paragraph / Sentence / Individual word.

At each level except the lowest one, individual words, the result of generalization of two expressions is a set of expressions. In such set, for each pair of expressions so that one is less general than other, the latter is eliminated. Generalization of two sets of expressions is a set of sets which are the results of pairwise generalization of these expressions.

Only a single generalization exists for a pair of words: if words are the same in the same form, the result is a node with this word in this form. To involve *word2vec* models (Mikolov et al., 2015), computing generalization of two different words, we use the following rule. If *subject1=subject2,* then *subject1^subject2 = <subject1, POS(subject1), 1>*. Otherwise, if they have the same part-of-speech, *subject1^subject2 =<*,POS(subject1), word2vecDistance(subject1^subject2)>*. If part-of-speech is different, generalization is an empty tuple. It cannot be further generalized.

For a pair of phrases, generalization includes all maximum ordered sets of generalization nodes for words in phrases so that the order of words is retained. In the following example

To buy digital camera today, on Monday

Digital camera was a good buy today, first Monday of the month

generalization is {*<JJ-digital, NN-camera> ,<NN- today, ADV,Monday>*} , where the generalization for noun phrases is followed by the generalization by adverbial phrase. Verb *buy* is excluded from both generalizations because it occurs in a different order in the above phrases. *Buy - digital - camera* is not a generalization phrase because *buy* occurs in different sequence with the other generalization nodes.

At the discourse level, rhetorical relations with elementary discourse units can be generalized as well. Only rhetorical relations of the same class (*presentation* relation, such as *antithesis, subject matter* relation, such as *condition, and multinuclear* relation, such as *list*) can be generalized. We use *N* for a nucleus or situations presented by this nucleus, and *S* for satellite or situations presented by this satellite. *Situations* are propositions, completed actions or actions in progress, and communicative actions and states (including *beliefs, desires, approve, explain, reconcile* and others). Hence we have the following expression for Rhetoric Structure Theory- based (RST, Marcu, 2000) generalization for two texts $text_1$ and $text_2$:

$text_1 \wedge text_2 = \cup_{i,j} (rstRelation_{1i} (\ldots,\ldots) \wedge rstRelation_{2j} (\ldots,\ldots))$,

where $I \in$ (RST relations in $text_1$), $j \in$ (*RST relations in* $text_2$). Further, for a pair of RST relations their generalization looks as follows: $rstRelation_1(N_1, S_1) \wedge rstRelation_2 (N_2, S_2) = (rstRelation_1 \wedge rstRelation_2)(N_1 \wedge N_2, S_1 \wedge S_2)$.

The texts in N_1, S_1 are subject to generalization as phrases. The rules for $rst_1 \wedge rst_2$ are as follows. If $relation\_type(rst_1) != relation\_type(rst_2)$ then similarity is empty. Otherwise, we generalize the signatures of rhetoric relations as sentences: $sentence(N_1, S_1) \wedge sentence (N_2, S_2)$ (Iruskieta et al., 2015).

To optimize the calculation of generalization score, we rely on a computational study which determined the POS weights to deliver the most accurate similarity measure between sentences possible (Galitsky et al., 2012). The problem was formulated as finding optimal weights for nouns, adjectives, verbs and their forms (such as gerund and past tense) such that the resultant search relevance is maximl. Search relevance was measured as a deviation in the order of search results from the best one for a given query (delivered by Google); current search order was determined based on

the score of generalization for the given set of POS weights (having other generalization parameters fixed). As a result of this optimization performed in (Galitsky et al., 2012), we obtained W_{NN} = 1.0, W_{JJ} = 0.32, W_{RB} = 0.71, W_{CD} = 0.64, W_{VB} = 0.83, W_{PRP} = 0.35 excluding common frequent verbs like *get/ take/set/put* for which $W_{VBcommon}$= 0.57. We also set that $W_{<POS,*>}$ =0.2 (different words but the same POS), and $W_{<*,word>}$ =0.3 (the same word but occurs as different POSs in two sentences).
Generalization score (or similarity between sentences $sent_1$, $sent_2$) then can be expressed as sum through phrases of the weighted sum through words $word_{sent1}$ and $word_{sent2}$
$score(sent_1, sent_2) = \sum_{\{NP, VP, \ldots\}} \sum W_{POS}\, word\_gen(word_{sent1}\, word_{sent2})$. The best generalization can then be defined as the one with the highest score. This way we define a generalization for phrases, sentences and paragraphs, as well as verb signatures from VerbNet, speech acts, communicative actions, as well rhetorical relations.

Result of the generalization can be further generalized with other parse trees or generalization. For a set of sentences, totality of generalizations forms a lattice: order on generalizations is set by the subsumption relation and generalization score (Khardon and Arias, 2006).

3 Question Answering Relevance via Generalization

To rank answers according to the relevance to a complex paragraph-sized question, paragraph-level generalization comes into play. It needs to involve a set of parse trees for each sentence of the paragraph and the paragraph-level discourse information. We refer to the sequence of parse trees plus a number of arcs for inter-sentence relations of the discourse type between the nodes for words as a *parse thicket* (Galitsky, 2017). It is a graph that includes parse trees for each sentence, as well as additional arcs for inter-sentence discourse relationships. We intend to demonstrate the richness of representation by parse thickets and the robustness of syntactic generalization operation on them for search relevance.

Our example is a web search query and its answers selected from the first page of a Google search. Although both answers A_1 and A_2 share very similar keywords, we show that using discourse information helps to differentiate them, relying on parse thicket representation and syntactic generalization operation with query Q.

Q: I am buying a foreclosed house. A bank offered me to waive inspection; however I am afraid I will not identify some problems in this property unless I call a specialist.
A_1: My wife and I are buying a foreclosure from a bank. In return for accepting a lower offer, they want me to waive the inspection. I prefer to let the bank know that I would not waive the inspection … Instead I would agree that I would pay the costs involved for a proper and thorough home inspection…
A_2: I am a foreclosure specialist in a bank which is subject to an inspection. FTC offered us to waive inspection if we can identify our potential problems with customers we lent money to buy their properties.

The reader can see that A_2 is totally irrelevant while A_1 is relevant.

We selected the first Google search result for the correct answer and composed a totally irrelevant answer with the same keywords to demonstrate the role of discourse-level generalization.

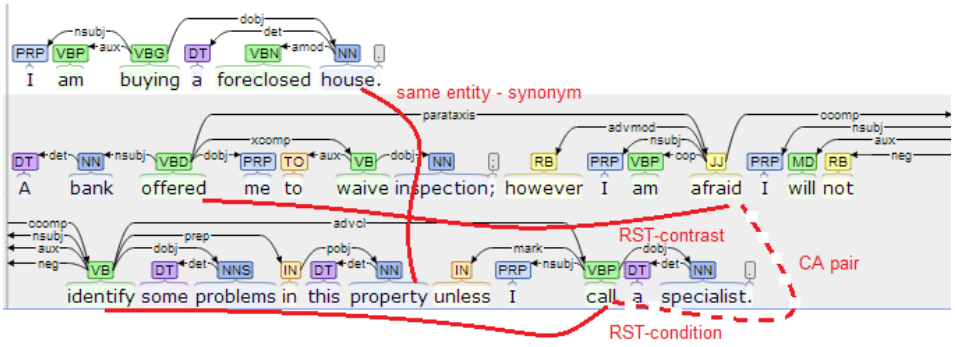


Fig. 1a parse thicket for question *Q*

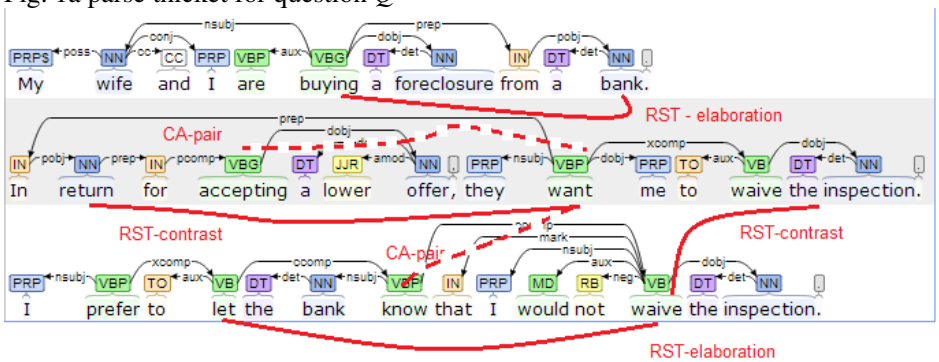


Fig. 1b Parse thicket for the valid answer A_1

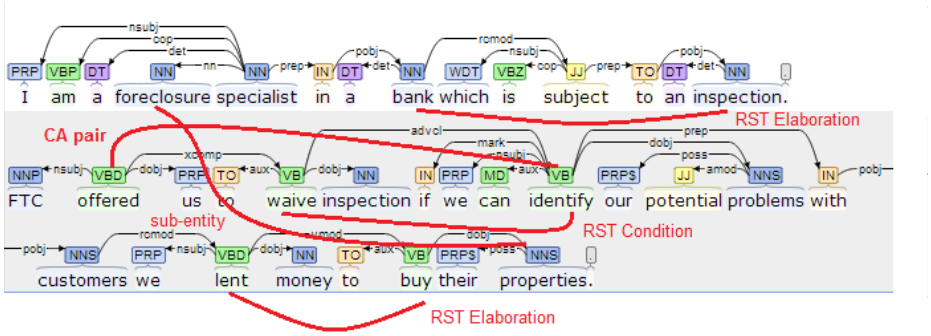


Fig. 1c: Parse thicket for the invalid answer A_2

The list of common keywords gives us a hint that both documents are about a relationship between the same entities, a *house, a buyer* and *a bank* in connection to a *foreclosure* and an *inspection*. However one can see that the relations between these entities in A_1 and A_2 are totally different. It is also obvious that something beyond the keyword statistics and n-gram analysis needs to be done to figure out the correspondence of the structure of these relations between *A1* and *Q*, and *A2* and *Q*.

Buy, foreclosure , house, bank, wave, inspection..

One can see that the key for the right answer here is rhetorical (RST) relation of *contrast*: *bank wants the inspection waved* but *the buyer does not*. Parse thicket generalization gives the detailed similarity picture that looks more complete than both the bag-of-words approach and pair-wise sentence generalization would. The similarity between *Q* and A_1 is expressed as a parse thicket expressed here as a list of phrases

[[NP [DT-a NN-bank], NP [NNS-problems], NP [NN*-property], NP [PRP-i]], VP [VB-* TO-to NN-inspection], VP [NN-bank VB-offered PRP-* TO-to VB-waive NN-inspection], VP [VB-* VB-identify NNS-problems IN-* NN*-property], VP [VB-* {phrStr=[], roles=[A, *, *], phrDescr=[]} DT-a NN-*]]]

And similarity with the invalid answer *A2* is expressed as a parse thicket formed as a list of phrases

[[NP [DT-a NN-bank], NP [PRP-i]], [VP [VB-* VB-buying DT-a], VP [VB-* PRP-me TO-to VB-waive NN-inspection], VP [VB-* {phrStr=[], roles=[], phrDescr=[]} PRP-i MD-* RB-not VB-* DT-* NN*-*],

The important phrases of the Q ^ A_1 similarity are *VP [NN-bank VB-offered PRP-* TO-to VB-waive NN-inspection], VP [VB-* VB-identify NNS-problems IN-* NN*-property]*,
which can be interpreted as a key topic of both Q and A_1*: bank* and not another entity should *offer to waive inspection*. This is what differentiates A_1 from A_2 (where these phrases are absent). Although *bank* and *problems* do not occur in the same sentences in Q and A1, they were linked by anaphora and RST relations. Without any kind of discourse analysis, it would be hard to verify whether the phrases containing *bank* and *problems* are related to each other. Notice that in A_2, problems are associated with *customers*, not *banks*, and different rhetoric relations from those common between Q and A_1 help us figure that out. Notice the semantic role attributes for verbs such as VB-* {phrStr=[], roles=[A, *, *], phrDescr=[]} in generalization result.

Parse thickets for Q, A_1 and A_2 are shown in Fig. 1a, 1b and 1c respectively. Notice the similarity in discourse structure of Q, A_1 and not in A_2*:* the RST-*contrast* arc. Also, there is a link for a pair of communicative actions for Q, A_1 (it is absent in A_2): *afraid-call* and *accept-want*.

4 Conclusions and Future Work

The generalization operation described earlier can be applied for the expanding of the training sets. It can multiply tail cases, make it more balanced, and eliminate noisy samples which cannot be generalized. We are planning to apply it in the number of the industrial linguistic applications with limited training datasets.

Acknowledgements

The article was prepared within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project '5-100'.

References

Galitsky, B., Gabor Dobrocsi, Josep Lluis de la Rosa, 2012. Inferring the semantic properties of sentences by mining syntactic parse trees. Data & Knowledge Engineering, V81-82 pp 21-45.

Galitsky, B. 2017. Matching parse thickets for open domain question answering. Data & Knowledge Engineering, v 107, pp. 24-50.

Robinson JA. A machine-oriented logic based on the resolution principle. Journal of the Association for Computing Machinery, 12:23-41, 1965.

Plotkin, GD. A note on inductive generalization. In B. Meltzer and D. Michie, editors, Machine Intelligence, volume 5, pages 153-163. Elsevier North-Holland, New York, 1970.

Khardon, Roni and Marta Arias. 2006. The subsumption lattice and query learning. Journal of Computer and System Sciences. v 72, Issue 1, February 2006, pp 72-94.

Mikolov, Tomas, Chen, Kai, Corrado; G.S., Dean; Jeffrey. 2015. Computing numeric representations of words in a high-dimensional space. US Patent 9,037,464, Google, Inc.

Marcu, D. 2000. Rhetorical Parsing of Unrestricted Texts. Computational Linguistics V 2 N3.

Ettinger, Allyson, Sudha Rao, Hal Daumé III, Emily Bender. Towards Linguistically Generalizable NLP Systems: A Workshop and Shared Task. 2017. EMNLP, Vancouver, Canada.

Kovalerchuk, B and Kovalerchuk, M. 2017. Toward Virtual Data Scientist with Visual Means. IJCNN.

T. Makhalova, D. Ilvovsky, B. Galitsky. Pattern Structures for News Clustering. FCA4AI@ IJCAI, 35-42

B. Ganter and S.O. Kuznetsov, Pattern Structures and Their Projections. ICCS 2001. Vol. 2120, pp. 129-142.

S.O. Kuznetsov, Fitting Pattern Structures to Knowledge Discovery in Big Data. ICFCA 2013. Vol. 7880, pp. 254–266, 2013.

S.O. Kuznetsov, Scalable Knowledge Discovery in Complex Data with Pattern Structures. PReMI'2013. Vol. 8251, pp. 30-41, 2013.

Truth and Justification in Knowledge Representation *

Andrei Rodin[1,2][0000−0002−3541−8867] and Serge Kovalyov[3][0000−0001−5707−5730]

[1] Institute of Philosophy RAS, 12/1 Goncharnaya Str., Moscow, 109240, Russian Federation
[2] HSE, 20 Myasnitskaya Ulitsa, Moscow, 101000, Russian Federation
avrodin@hse.ru
https://www.hse.ru/org/persons/218712080
[3] ICS RAS, 65 Profsoyuznaya street, Moscow 117997, Russian Federation
kovalyov@sibnet.ru https://www.ipu.ru/node/16660

Abstract. While the traditional philosophical epistemology stresses the importance of distinguishing knowledge from true beliefs, the formalisation of this distinction with standard logical means turns out to be problematic. In Knowledge Representation (KR) as a Computer Science discipline this crucial distinction is largely neglected. A practical consequence of this neglect is that the existing KR systems store and communicate knowledge that cannot be verified and justified by users of these systems without external means. Information obtained from such systems does not qualify as knowledge in the sense of philosophical epistemology.
Recent advances in the research area at the crossroad of the computational mathematical logic, formal epistemology and computer science open new perspectives for an effective computational realisation of justificatory procedures in KR. After exposing the problem of justification in logic, epistemology and KR, we sketch a novel framework for representing knowledge along with relevant justificatory procedures, which is based on the Homotopy Type theory (HoTT). This formal framework supports representation of both propositional knowledge, aka knowledge-that, and non-propositional knowledge, aka knowledge-how or procedural knowledge. The default proof-theoretic semantics of HoTT allows for combining the two sorts of represented knowledge at the formal level by interpreting all permissible constructions as justification terms (witnesses) of associated propositions.

Keywords: Knowledge Representation · Justification · Homotopy Type theory

1 Concept of Knowledge

1.1 Knowledge according to the Philosophical Epistemology

JTB and Gettier Problem The current philosophical discussion on the concept of knowledge focuses on the so-called JTB theory of knowledge according

* Supported by RFBR grant 19-011-00799.

to which knowledge is Justified True Belief. The JTB theory dates back to Plato and, in modern terms, states that subject S knows that p (where p is a proposition) just in case the following three conditions are satisfied [1]:

1. p is true
2. S believes that p
3. S is justified in believing that p.

Leaving the psychological concept of belief aside of our present analysis we would like to stress the following features of JTB theory:

- JTB identifies knowledge with knowledge of certain proposition or propositions; this type of knowledge is conventionally referred to as propositional knowledge aka knowledge-that.
- JTB assumes that the truth-value of a given proposition is determined wholly independently of one's knowledge of this proposition. Such an account of truth has a long tradition in logic and has been strongly defended, among other people, by Gottlob Frege. We shall shortly show that this conception of truth is not commonly accepted in the philosophical logic.
- According to JTB, a true belief, i.e., one's belief in certain true proposition, by itself does not constitute knowledge. A missing element is justification. Assuming that a mathematical proof is a special form of justification, for a motivating example think of Bob who is able to state the Pythagorean theorem (provided she understands its meaning and believes it is true) and Alice who is also able to prove it. In terms of JTB theory Alice *knows* the theorem but Bob doesn't. What is at stake here is not the linguistic meaning of "know" but the difference between the two sorts of epistemic states, viz. knowledge and (true) belief (or however one may prefer to call them).

A major part of the mainstream discussion on and about JTB concerns the so-called Gettier Problems. In 1963 Edmund Gettier published a highly influential paper [2] where he showed using some linguistic examples that the concept of justification involved into JTB is very problematic: a subject can be compelled to belief that p, where p is true proposition, by certain reason r, which she relates to p by a mere mistake; allegedly such "false reasons" cannot be ruled out by the JTB theory, so this theory is at best incomplete and at worst wholly wrong. Without being able to discuss here Gettier-style epistemological problems systematically we would like to express our general take on this issue: in our view, the core problem here is that the concept of justification unlike that of truth is not adequately accounted for by standard logical tools. On the contrary to a popular opinion it also concerns mathematical proofs. However during last decades there was a significant progress towards this goal some elements of which are described in what follows.

Since JTB accounts only for the propositional knowledge certain authors argue that it doesn't cover the concept of knowledge in its full extent leaving aside an irreducibly *procedural* knowledge aka knowledge-how. We endorse this view and remark after Ryle [3] that knowing how to make logical inferences and otherwise justify one's beliefs is (irreducibly) procedural rather than propositional

in its character. Another challenge for JTB comes from the constructive tradition in logic, which tightly relates truth and knowledge by identifying truth of proposition with the existence of its proof (evidence). From the constructive point of view the tripartition of knowledge into (i) true proposition, (ii) one's belief in this proposition and (iii) justification of this belief is hardly tenable because here truth of a given proposition requires its justification (proof) in some form at the first place. Notice that even if this constructive approach is incompatible with JTB in its usual form, it shares with JTB the notion according to which justification is a necessary element of knowledge. In fact the constructive approach takes justification to be even more important by making it constitutive for truth and logic itself. M. Cohen and E. Nagel express this view on logic when they describe its purpose, in full generality, as the "determination of the best available evidence" [4].

1.2 Knowledge in Computer Science and Information Technology

Knowledge Representation (KR) sometimes also referred to as Knowledge Representation and Reasoning is an established discipline in Computer Science and a devoping information technology. Obviously one should not expect to find in KR literature a thorough analysis of the knowledge concept, which can be more appropriate in the philosophical literature. Nevertheless authors of some monographs and textbooks in KR provide informal descriptions of basic concepts of the discipline including that of knowledge and reasoning [5],[6],[7]. Remarkably, none of such descriptions found by us in the CS literature mentions the standard epistemological condition according to which knowledge needs to be justified.

This observation squares with another one. In 1980-ies the philosophical term "ontology" began its independent life in CS and since 1995 the latest [8] has been used systematically as a standard technical term and concept in the KR design and research. In philosophy the term "ontology" refers to the problematic area of research and reflection that concerns, to use Aristotle's famous word, the *being qua being* or, in more modern terms, general features of all entities in their mere capacity of being existent: it includes classifications of entities into different sorts (e.g., objects, events and their properties) and the like. Ontologies used in KR are computationally implemented formal semantic frameworks for representing objects and their mutual relations; knowledge represented in a KR system refers to these objects and relations as its subject-matter, i.e., to what this knowledge is "about".

[4] Here is the full quote:

> [T]he constant and universal feature of science is its general method, which consists in the persisting search for truth, constantly asking: Is it so? To what extent is it so? Why is it so? [...] And this can be seen on reflection to be the demand for the best available evidence, the determination of which we call logic. Scientific method is thus the persistent application of logic as the common feature of all reasoned knowledge [4, p. 192]

Despite the fact that KR ontologies lose at some extent their philosophical origins, the difference between the philosophical and the CS concepts of ontology is not dramatic. *Formal ontology*, which is a philosophical ontology developed with a support of formal logical methods, can be seen as a middle ground that links the traditional philosophical ontology, on the one hand, and the technical concept of ontology used in KR, on the other hand.

What is puzzling here in eyes of a philosopher is the following. A philosophical discipline that covers problems concerning knowledge is called epistemology but not ontology. Just like ontology epistemology is developed, in part, with a support of formal methods; this approach is known as *formal epistemology*. Yet, the CS discipline that essentially involves the concept of knowledge, viz. KR, for some reason makes use of ontology but not of epistemology.

We don't assume here that CS or any other engineering discipline must respect traditional philosophical distinctions when it borrows philosophical terms and concepts and then modifies them for its own use. However we claim that the above observations point to a real problem, which has a practical dimension. The issue of reliability of information available via electronic communications is widely discussed in special and general literature and since recently is also recognised as an important social and even political problem [9]. The existing data verification technologies are designed for serving developers and administrators of KR systems rather than its regular users, and for this reason don't fully address this problem. In order to make a piece of information obtained by a user of KR system reliable in eyes of this very user (as this is required by the JTB conception of knowledge and by any other conception that takes the issue of justification seriously) a supporting evidence needs to be available to the user herself. We assume here that this evidence also needs to be specific and not reduce to a general assurance that the given KR system is reliable.

A part of the problem, as we see it, is that the standard logical tools such as the first-order Classical logic along with its usual philosophical underpinning leave the epistemic concept of justification aside. The philosophical conceptions of truth and logical reasoning that underline this notion of logic prioritise ontological aspects with respect to epistemological ones. Correspondingly, theoretical prototypes of KR systems, which use this standard logical and semantic framework, essentially use ontologies but don't support the epistemic procedure of justification. If Sundolm is right that epistemic considerations have been systematically neglected in the mainstream logical research until very recently [10], it is a little surprise that they have been also neglected in CS. In the next Section we elaborate on this point providing more details and then propose a remedy.

2 Model-theoretic and Proof-theoretic Semantics

The standard notion of axiomatic theory that stems from Hilbert assumes that:

- A theory is set T of formulas that are interpreted as true statements; such interpretations of formulas are called models of the given theory;
- A theory has a subset of formulas $A \subset T$ called axioms of the given theory;
- A theory comprises set R of syntactic rules, which, in particular, regulate derivations of new T-formulas from some given T-formulas. T-derivations preserve truth in the sense that given any model m of T, they derive from true sentences only true sentences (soundness). T comprises all formulas T-derivable from its axioms (deductive closure).
- T-formulas, which are T-derivable from the axioms (other than the axioms) are called theorems of T. A derivation of theorem from axioms (and by extension also from some intermediate theorems) is called a proof of a given theorem. The standard notation for the derivability of theorem B from axioms $A_1, \ldots, A_n$ in theory T is as follows: $A_1, \ldots, A_n \vdash_T B$.

This familiar scheme involves at least one epistemic term, namely, "proof". However as convincingly, in our view, argues Prawitz the bold identification of proofs with syntactic derivations is not justified [11]. In order to explain the argument we need the following formal notion of *logical consequence relation* due to Tarski [12]:

Definition 1 *T-formula B is called a* logical consequence *of T-formulas $A_1, \ldots, A_n$, in symbols $A_1, \ldots, A_n \models_T B$ just is case every interpretation m that interprets $A_1, \ldots, A_n$ as true sentences also interprets B as true sentence.*

Since T is sound (with respect to some fixed class of its interpretations), every T-theorem B derived from T-axioms $A_1, \ldots, A_n$ is a logical consequence of these axioms. *Prima facie* this observation justifies the idea that a formal derivation of B represents its logical inference from T-axioms, which qualifies as its T-proof. Prawitz argues to the contrary. Even if it is the case that a given syntactic derivation faithfully represents a truth-preserving logical inference, nothing guarantees in this setting that the same symbolic representation allows one to *see* that truth is preserved. A further problem concerns the specific Hilbertian notion of axiom and the related Tarskian notion of *truth-in-a-model*, which has little to do with the traditional notion of axiom as a self-evident truth. The concept of evidence is wholly alien to this approach and ruled out as psychological and hence logically irrelevant. This makes a sharp contrast with the aforementioned Cohen&Nagel's conception of logic where the notion of evidence has a central role [4]. What is an epistemic value, if any, of a formal proof in the above technical sense remains unclear.

At the same time the above logico-semantic framework can be straightforwardly related to ontology via the following principle known as the *truthmaker realism* (TMR):

Given a true statement there exists an entity (or entities) that make(s) this statement true. [13]

Once one accepts TMR the notion of formal ontology readily suggests itself as a useful formal semantic tool, which helps one to supplement, and in many applications even to replace, the talk of models and interpretations by talks about some familiar entities that a given theory is supposed to account for. The Tarski-style formal semantics helps to make this ontological talk formal and rigorous. This is a pragmatic reason to accept some form of TMR and the notion of formal ontology, which may convince even those people, including KR developers, who are not interested in traditional philosophical debates about being and existence. One does not need to explore deep philosophical questions about being in order to use formal ontologies as semantic tools. This explains why the notion of formal ontology became useful and popular in the AI research.

However, as we have already stressed, the neglect of epistemic considerations in the foundations of the above logico-semantic framework and, more specifically, the lack of satisfactory formal treatment of justification, rises a problem, which is not only theoretical but also practical. Once the theoretical and practical significance of justification is recognised, it becomes clear that the standard logical and semantical tools are not sufficient for developing theoretical prototypes of reliable KR systems.

In the philosophical and mathematical logic this epistemological problem is well recognised and understood by a part of the professional community. There is presently a number of tentative solutions on the market. A systematic formal treatment of Justification Logic with explicit justificatory terms is given in the new monograph by S. Artemov and M. Fitting [14]. A variety of approaches that attempt to supplement or replace the standard model-theoretic logical semantics (MTS) outlined above by some version of alternative epistemically relevant semantic is now grouped under the header of *proof-theoretic semantics* (PTS) [15]. It is remarkable that many versions of PTS are more "computer-friendly", i.e., more apt for computer implementation, than their MTS analogues because they give semantic values directly to syntactic rules and procedures rather than only to formulas. A systematic overview of this actively developing area of research is out of the scope of this paper. In the next Section we only briefly describe a formal theory that belongs to the PTS family (albeit arguably goes beyond PTS in some essential aspects) and propose it to the role of novel formal semantic framework for KR.

3 Homotopy Type theory as KR framework

3.1 MLTT, HoTT and Their Proof-Theoretic Semantics

Homotopy Type theory (HoTT) is a growing family of type theories with dependent types, which are interpreted (more or less formally) in terms of Homotopy theory, which is a part of Algebraic Topology[5]. Such homotopical interpretations of type theories were discovered independently by Steven Awodey and Vladimir

[5] The exposition of MLTT/HoTT found in this subsection reuses some materials published in [18].

Voevodsky in mid 2000-ies. We consider here the standard HoTT presented in [17], which uses the syntax of Martin-Löf's Type theory (MLTT) [16] extended with the single Univalence Axiom, which is out of the scope of the present discussion. This version of HoTT preserves the core proof-theoretic semantics of the original MLTT and extends it with a new homotopy semantics. We analyse the relationships between the original MLTT semantics and the HoTT semantics and attempt to make their combination coherent.

MLTT is a rule-based formal system that comprises no axiom. Its basic formulas are called *judgements* and interpreted accordingly. MLTT comprises four basic forms of *judgements*.

(i) $A : TYPE$;
(ii) $A \equiv_{TYPE} B$;
(iii) $a : A$;
(iv) $a \equiv_A a'$

In words (i) says that A is a type, (ii) that types A and B are the same, (iii) that a is a term of type A and (iv) that a and a' are the same term of type A. We now leave (i) and (ii) aside and provide more details on (iii) and (iv).

Martin-Löf offers four different informal readings of (iii) [16, p. 5]:

1. a is an element of set A
2. a is a proof (construction) of proposition A ("propositions-as-types")
3. a is a method of fulfilling (realizing) the intention (expectation) A
4. a is a method of solving the problem (doing the task) A (BHK semantics)

The author argues that these interpretations of judgement form (iii) not only share a logical form but also are closely conceptually related despite of their different linguistic appearance.

Let us now turn to judgement form (iv). It says that terms a, a', both of the same type A, are equal. This equality is called *judgemental* or *definitional* and does *not* qualify as a proposition; the corresponding *propositional* equality writes as $a =_A a'$ and counts as a type on its own ($a =_A a' : TYPE$) called an *identity type*. In accordance to reading (2) of judgement form (iii) a term of identity type is understood as a proof (also called a witness or evidence) of the corresponding proposition. MLTT validates the rule according to which a judgemental equality entails the corresponding propositional equality:

$$\frac{a \equiv_A a'}{refl_a : a =_A a'}$$

where $refl_a$ is the canonical proof of proposition $a =_A a'$.

The *extensional* version of MLTT also validates the converse rule called the *equality reflection rule*:

$$\frac{p : a =_A a'}{a \equiv_A a'}$$

HoTT draws on an *intensional* version of MLTT that does not use such a principle and allows for multiple proofs of the same propositional equality.

Let now p, q be two judgmentally different proofs of proposition saying that two terms of a given type are equal:

$$p, q : P =_T Q$$

it may be the case that p, q, in their turn, are propositionally equal, and that there are two judgmentally different proofs p', q' of this fact:

$$p', q' : p =_{P=_T Q} q$$

This and similar multi-layer syntactic constructions in MLTT can be continued unlimitedly. Before the rise of HoTT it was not clear that this syntactic feature of the intensional MLTT can be significant from a semantic point of view. However it became the key point of the homotopical interpretation of this syntax. Under this interpretation

- types and their terms are interpreted, correspondingly, as spaces and their points;
- identity proofs of form $p, q : P =_T Q$ are interpreted as paths between points P, Q of space T;
- identity proofs of the second level of form $p', q' : p =_{P=_T Q} q$ are interpreted as homotopies between paths p, q;
- all higher identity proofs are interpreted as higher homotopies;

Recall that *path* p between points P, Q of topological space T is continuous map $p : [0, 1] \to T$ such that $p(0) = P$ and $p(1) = Q$. Intuitively a path can be thought of as a trajectory of moving test point where the real interval [0,1] represents time. In a more abstract presentation the real unit interval [0,1] is replaced by abstract unit object I. *Homotopy* h between paths p, q is continuous map $h : [0, 1]^2 \to T$ such that $h(t, 0) = p(t)$ and $h(t, 1) = q(t)$; intuitively it can be thought of as a "path between paths" or a continuous transformation of path p into path q. Higher homotopies are defined similarly. For a modern introduction into the basic Homotopy theory see [19].

The homotopical interpretation makes the complex structure of identity types in the intensional MLTT surveyable and suggests a revision of the original semantics of MLTT by distinguishing between propositional and non-propositional types on the syntactic level. According to this new point of view not every type can be interpreted either as a proposition or as a set but each of these two interpretations is admissible only for types of appropriate sorts. More precisely, consider the following

Definition 2 *Space aka homotopy type S is called* contractible *or space (type) of h-level (-2) when there is point $p : S$ connected by a path with each point $x : A$ in such a way that all these paths are homotopic (i.e., there exists a homotopy between any two such paths).*

Definition 3 *We say that S is a space of h-level $n+1$ if for all its points x, y path spaces $x =_S y$ are of h-level n.*

These definitions gives rise to the following stratification of types/spaces in HoTT by their h-levels:

- h-level (-2): single point pt;
- h-level (-1): the empty space $\emptyset$ and the point pt: truth-values aka (mere) propositions;
- h-level 0: sets aka discrete point spaces: comprise no non-contractible paths;
- h-level 1: flat path groupoids : comprise paths but no non-contractible surfaces;
- h-level 2: 2-groupoids : comprise paths and surfaces but no non-contractible volumes;
- ...
- h-level ω: ω-groupoids.

Space S_n of h-level n can be transformed into a space $[S]_k$ of h-level $k < l$ via its *k-truncation*, which can be informally described as a forced identification of all homotopies (paths) of all levels higher than k. In particular, the (-1)-truncation $[S]_{-1}$ of any given space S brings point pt when S is not empty and brings the empty space $\emptyset$ otherwise.

The notion of truncation allows for interpreting type $[S]_{-1}$ as a proposition and the original type S as a (h-stratified) space of proofs of this proposition: $[S]_{-1}$ is true when it has a proof in S and is false otherwise. Assuming that the scope of logic restricts to propositional types one can now describe higher types and their terms as being extra-logical. However the homotopic semantics of the extra-logical terms still qualifies as proof-theoretic because such terms serve as proof terms of certain propositions.

3.2 How to use HoTT for KR purposes

If different terms of the same type are not distinguished then HoTT is functioning as a constructive propositional logic with explicit proof terms, which in this case can be also called internalised truth-values. If the (in)equalities of terms are taken into account only up to the set level (which means that distinctions between different paths between the same terms, i.e., between "different ways of being equal", are ignored) then HoTT functions as a constructive first-order calculus with internalised (constructive) sets that already provides more information about its proof terms. These sets are constructively internalised in the sense that they are represented here with syntactic constructions available in HoTT itself rather than introduced with a help of some external meta-theoretical tools as this is done in case of standard Tarski semantics for the Classical first-order logic.

This feature alone demonstrates a potential of HoTT as a representational framework: it supports representation of propositions along with objects that

those propositions are "about"; the same terms can be also described as truth-makers of their base propositions, their evidences or their proofs. Accordingly, HoTT represents a propositional knowledge (since the true represented propositions are evidenced) along with an associated procedural knowledge, viz., the knowledge of how to construct for the given proposition its evidences aka proofs. Such a justificatory procedure for propositional knowledge has its formal dual in the form of verification of the corresponding procedural knowledge. In this case the epistemic goal is not to justify a propositional belief but to assure that an accomplished construction has some required properties. Think of technological processes which certain desired outcomes, which needs to be checked and verified. Since this difference in epistemic goals does not affect the basic semantics of HoTT, our proposed approach applies to both these sorts of tasks.

Higher levels of the homotopy ladder provide more expressive power for representing objects and spaces where these objects live. The (flat) groupoid spaces (h-level 1) already allow for representing certain non-trivial topological features of the base spaces. Leaving for another occasion a study of possible applications of topological concepts in KR we would like to stress here its intuitive appeal. This is not a minor issue when we are talking about possible ways to justify knowledge obtained via a KR system, which is supposed to be available to a regular user. A HoTT-based approach has been already successfully used for an automated verification of non-trivial mathematical proofs [21]. An advantage of this approach over other approaches in the automated proof verification is that the homotopical interpretation allows a mathematician to express her reasoning with a commented program code or a pseudo-code without giving up the usual intuitive support of this reasoning. This specific feature of HoTT might be helpful for designing a format for human-readable evidences or certificates that a hypothetical KR system could produce in order to justify the supplied knowledge in eyes of its user. A toy example of HoTT-based representation used outside the pure mathematics is found in [20].

We summarise our explanation of relevance and possible advantages of using HoTT as a formal KR framework that supports justification as follows:

1. HoTT admits the constructive epistemically-laden proof-theoretic semantics intended by Martin-Löf's Type for MLTT (in a slightly modified form).
2. The cumulative h-hierarchy of types made explicit via the homotopical interpretation supports the distinction between propositional, set-level and higher-level types. This distinctive feature of HoTT supports formal constructive representation of objects (of various levels) and propositions "about" these objects within the same framework. Each such object serves as a witness/truthmaker for proposition obtained via the propositional truncation of type where the given object belongs.
3. HoTT comprises a system of formal rules, which are interpreted as logical rules at the propositional h-level and as rules for object-construction at all higher levels. This feature of HoTT supports representation various extra-logical procedures (such as material technological procedures) keeping track

of the corresponding logical procedures at the propositional level of representation.
4. HoTT/MLTT is computer-friendly, i.e., computationally implementable. Fragments of HoTT/MLTT have been implemented in proof-assistant Coq, program languages AGDA, LEAN and some other products.
5. HoTT-constructions admit intuitive spatial (homotopical) interpretations that may be used for facilitating human-computer interactions.

4 Conclusion

During the last decade KR technologies have been enriched with approaches based on the Big Data analysis, Machine Learning and artificial Neural Networks. According to a radical opinion, these new approaches make more traditional logical approaches based on the explicit representation of facts and rules hopelessly outdated and irrelevant. We disagree. Because of their possible unpredictable behaviour [22] Neural Networks and other tools of Big Data analysis can significantly enrich but not replace logical approaches and logical tools in KR.

At the same time we agree that standard logical architectures and formal ontologies, which are presently used in KR, don't provide a sufficient theoretical background for KR because they have no epistemological content. In this paper we explained the relevance of epistemological considerations in logic and KR and then pointed to some recent advances in mathematical logic, more specifically discussing the Homotopy Type theory, that may allow to use logical approaches in KR more effectively.

References

1. Ichikawa, J.J. and Steup, M., "The Analysis of Knowledge", The Stanford Encyclopedia of Philosophy (Summer 2018 Edition), Edward N. Zalta (ed.), URL = https://plato.stanford.edu/archives/sum2018/entries/knowledge-analysis/
2. Gettier, E. L., "Is Justified True Belief Knowledge?", Analysis, 23(6): 121123 (1963), https://doi.org/10.2307/3326922
3. Ryle, G., "Knowing How and Knowing That: The Presidential Address", Proceedings of the Aristotelian Society, New Series, 46: 1-16 (1945-46)
4. Nagel, E. and Cohen, M.R., An Introduction to Logic and Scientific Method, Routledge, 1934
5. Jakus, G. et al., Concepts, Ontologies and Knowledge Representation, Springer 2013
6. Abraham, A. and Grosan, C., Intelligent Systems : A Modern Approach, Springer 2011
7. Lakemeyer, G. and Nebel, B. (Eds.) , Foundations of Knowledge Representation and Reasoning, Springer 1994
8. Gruber, Th.R., "Toward principles for the design of ontologies used for knowledge sharing?", International Journal of Human-Computer Studies, 43(5-6):907-928 (1995)
9. Fuller, S., Post-Truth: Knowledge as a Power Game, Cambridge University Press, 2018

10. Sundholm, G. “The Neglect of Epistemic Considerations in Logic: the Case of Epistemic Assumptions”, Topoi 38 (3): 551-559 (2018) https://doi.org/10.1007/s11245-017-9534-0 (Open Access)
11. Prawitz, D., “On the Idea of the General Proof Theory”, Synthese, 27(1-2): 63-77 (1974)
12. Tarski, A., “On the Concept of Logical Consequence”, In: Logic, Semantics, Metamathematics, Hackett Publ., 1983, pp. 409-420
13. Smith, B. “Truthmaker realism”, Australasian Journal of Philosophy, 77 (3): 274-291 (1999)
14. Artemov, S. and Fitting, Justification Logic: Reasoning with Reasons, Cambridge University Press, 2019
15. Piecha, Th. and Schroeder-Heister, P. (Eds.), Advances in Proof-Theoretic Semantics, Springer 2015
16. Martin-Löf, P., Intuitionistic Type Theory, BIBLIOPOLIS, 1984
17. Univalent Foundations Program: Homotopy Type Theory, IAS Princeton, 2013 URL = https://homotopytypetheory.org/book/ (Open Access)
18. Rodin, A., “Models of HoTT and the Constructive View of Theories”, In: Centrone, S., Kant, D. and Sarikaya, D. (Eds.) Reflections on the Foundations of Mathematics: Univalent Foundations, Set Theory and General Thoughts, Springer 2019, pp. 189-220
19. Strom, J., Modern Classical Homotopy Theory, American Mathematical Society 2011
20. Rodin, A., “Venus Homotopically”, IfCoLog Journal of Logics and their Applications, 4(4): 1427-1446 (2017, open access)
21. Grayson, D.R., “An introduction to univalent foundations for mathematicians”, Bulletin of the American Mathematical Society 55 (2017), https://doi.org/10.1090/bull/1616.
22. Szegedy, C. et al.,“ Intriguing Properties of Neural Networks”, CoRR abs/1312.6199 (2013)

FCA-based Approach to Machine Learning*

Dmitry V. Vinogradov[1,2]

[1] Federal Research Center for Computer Science and Control,
Russian Academy of Science, Moscow 119333, Russia
vinogradov.d.w@gmail.com
[2] Russian State University for Humanities, Intelligent Robotics Laboratory,
Moscow 125993, Russia
WWW home page: http://isdwiki.rsuh.ru/index.php/User:Vinogradov.Dmitry

Abstract. The main result of the paper provides a lower bound on sufficient number of randomly generated formal concepts to correctly predict all important positive test examples with given confidence level. The technique coincides with modern approach to the famous theorem of V.N. Vapnik and A.Ya. Chervonenkis. However the situation is dual to the classical one: in our case test examples correspond to fixed subsets and probabilistically generated formal concepts must fall into selected areas of sufficient large volume.

Keywords: formal context, formal concept, Boolean hypercube, lower half-space, prediction, confidence

1 Introduction

Formal Concept Analysis (FCA) [1] is a popular means based on lattice theory for formalizing methods of data analysis in case of small samples.

Applicability of FCA to Big Data has several obstacles:

- Exponentially large number of hypotheses with respect to size of the initial formal context in the worst case.
- Many problems of FCA belong to famous classes of NP- and $\#P$-complete problems [3].
- There is a positive probability of "accidental" concepts appearance that correspond of overfitting phenomenon [7].

The paper [6] introduces the Markov chain approach to probabilistic generation of formal concepts (so-called VKF-method). The computer VKF-system uses the coupling Markov chain to generate random sample of concepts. Each run of this chain terminates with probability 1. Since each hypothesis (formal concept) is generated by independent run of the Markov chain, the system makes the induction step in parallel by several threads. Finally the system predicts target class of each test example by the analogy reasoning.

* Partially supported by RFBR grant 17-07-00539A

The key question of the approach is how to determine sufficient number of hypotheses to predict target class with given level of confidence. The paper proposes an answer to this question.

Used technique mostly coincides with modern approach to the famous theorem of V.N. Vapnik and A.Ya. Chervonenkis. However the situation is dual to the classical one: in our case test examples correspond to fixed subsets and probabilistically generated formal concepts must fall into selected areas of sufficient large volume. The general approach of Vapnik-Chervonenkis uses the "Occam razor" principle where no assumption on selected hypothesis made except to its correctness on all training examples. Hence a hypothesis coincides with area of objects space. To reject a bad hypothesis is needed to randomly pick training objects from the corresponding subset.

2 Background

2.1 Basic definitions and facts of FCA

Here we recall some basic definitions and facts of Formal Concept Analysis (FCA) [1].

A **(finite) context** is a triple (G, M, I) where G and M are finite sets and $I \subseteq G \times M$. The elements of G and M are called **objects** and **attributes**, respectively. As usual, we write gIm instead of $\langle g, m\rangle \in I$ to denote that object g has attribute m.

For $A \subseteq G$ and $B \subseteq M$, define

$$A' = \{m \in M | \forall g \in A(gIm)\}, \tag{1}$$

$$B' = \{g \in G | \forall m \in B(gIm)\}; \tag{2}$$

so A' is the set of attributes common to all the objects in A and B' is the set of objects possesing all the attributes in B. The maps $(\cdot)' : A \mapsto A'$ and $(\cdot)' : B \mapsto B'$ are called **derivation operators** (**polars**) of the context (G, M, I).

A **concept** of the context (G, M, I) is defined to be a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The first component A of the concept (A, B) is called the **extent** of the concept, and the second component B is called its **intent**. The set of all concepts of the context (G, M, I) is denoted by $\mathbf{B}(G, M, I)$.

Let (G, M, I) be a context. For concepts (A_1, B_1) and (A_2, B_2) in $\mathbf{B}(G, M, I)$ we write $(A_1, B_1) \leq (A_2, B_2)$, if $A_1 \subseteq A_2$. The relation $\leq$ is a **partial order** on $\mathbf{B}(G, M, I)$.

A subset $A \subseteq G$ is the extent of some concept if and only if $A'' = A$ in which case the unique concept of which A is the extent is (A, A'). Similarly, a subset B of M is the intent of some concept if and only if $B'' = B$ and then the unique concept with intent B is (B', B).

Proposition 1. *[1] Let (G, M, I) be a context. Then $(\boldsymbol{B}(G, M, I), \leq)$ is a lattice with join and meet given by*

$$\bigvee_{j \in J} (A_j, B_j) = ((\bigcup_{j \in J} A_j)'', \bigcap_{j \in J} B_j), \tag{3}$$

$$\bigwedge_{j \in J} (A_j, B_j) = (\bigcap_{j \in J} A_j, (\bigcup_{j \in J} B_j)''); \tag{4}$$

□

Corollary 1. *For context (G, M, I) the lattice $(\boldsymbol{B}(G, M, I), \leq)$ has (M', M) as the bottom element and (G, G') as the top element. In other words, for all $(A, B) \in \boldsymbol{B}(G, M, I)$ the following inequalities hold:*

$$(M', M) \leq (A, B) \leq (G, G'). \tag{5}$$

□

Definition 1. *For $(A, B) \in \boldsymbol{B}(G, M, I)$, $g \in G$, and $m \in M$ define*

$$CbO((A, B), g) = ((A \cup \{g\})'', B \cap \{g\}'), \tag{6}$$

$$CbO((A, B), m) = (A \cap \{m\}', (B \cup \{m\})''). \tag{7}$$

so $CbO((A, B), g)$ is equal to $(A, B) \vee (\{g\}'', \{g\}')$ and $CbO((A, B), m)$ is equal to $(A, B) \wedge (\{m\}', \{m\}'')$.

We call these operations CbO because the first one is used in Close-by-One (CbO) Algorithm to generate all the elements of $\mathbf{B}(G, M, I)$, see [2] for details.

Useful properties of introduced operations are summarized in the following Lemmas.

Lemma 1. *Let (G, M, I) be a context, $(A, B) \in \boldsymbol{B}(G, M, I)$, $g \in G$, and $m \in M$. Then*

$$g \in A \Rightarrow CbO((A, B), g) = (A, B), \tag{8}$$

$$m \in B \Rightarrow CbO((A, B), m) = (A, B), \tag{9}$$

$$g \notin A \Rightarrow (A, B) < CbO((A, B), g), \tag{10}$$

$$m \notin B \Rightarrow CbO((A, B), m) < (A, B). \tag{11}$$

Lemma 2. *Let (G, M, I) be a context, $(A_1, B_1), (A_2, B_2) \in \boldsymbol{B}(G, M, I)$, $g \in G$, and $m \in M$. Then*

$$(A_1, B_1) \leq (A_2, B_2) \Rightarrow CbO((A_1, B_1), g) \leq CbO((A_2, B_2), g), \tag{12}$$

$$(A_1, B_1) \leq (A_2, B_2) \Rightarrow CbO((A_1, B_1), m) \leq CbO((A_2, B_2), m). \tag{13}$$

Now we represent the coupling Markov chain algorithm that is a core of probabilistic approach to machine learning based on FCA (VKF-method).

Data: context (G, M, I), external function $CbO(\ ,\ )$
Result: random concept $(A, B) \in \mathbf{B}(G, M, I)$
$X := G \sqcup M$; $(A, B) := (M', M)$; $(C, D) = (G, G')$;
while $((A \neq C) \vee (B \neq D))$ **do**
  select random element $x \in X$;
  $(A, B) := CbO((A, B), x)$; $(C, D) := CbO((C, D), x)$;
end

Algorithm 1: Coupling Markov chain

The order on two concepts $(A, B) \leq (C, D)$ at any intermediate step of the while loop of Algorithm 1 follows from Lemma 2.

2.2 Probabilistic algorithms for FCA-based machine learning

Now we represent the general scheme of machine learning based on FCA (VKF-method). The reader can learn the classical deterministic FCA-based approach to machine learning from Kuznetsov [4]. Our technique uses probabilistic Algorithm 1 for computing a random subset of formal concepts.

As usual, there are two sets of objects called the training and test samples, respectively.

From positive examples of the training sample the program generates a formal context (G^+, M, I). The negative examples form the set G^- of counter-examples (**obstacles**).

Set G^τ of examples to predict the target class contains all test objects.

After that the program applies the coupling Markov chain algorithm 1 to generate a random formal concept $(A, B) \in \mathbf{B}(G^+, M, I)$. The program saves the concept (A, B), if there is no obstacle $o \in G^-$ such that $B \subseteq o'$.

Data: number N of concepts to generate
Result: random sample S of formal concepts without obstacles
$G^+ :=$ (+)-examples, $M :=$ attributes; $I \subseteq G^+ \times M$ is a formal context for (+)-examples;
$G^- :=$ (-)-examples; $S := \emptyset$; $i := 0$;
while $(i < N)$ **do**
  Generate concept $\langle A, B\rangle$ by Algorithm 1; $hasObstacle :=$ **false**;
  for $(o \in G^-)$ **do**
    if $(B \subseteq o')$ **then**
      $hasObstacle :=$ **true**;
    end
  end
  if $(hasObstacle =$ ***false***$)$ **then**
    $S := S \cup \{\langle A, B\rangle\}$;
    $i := i + 1$;
  end
end

Algorithm 2: Inductive generalization

Condition ($B \subseteq o'$) of Algorithm 2 means the inclusion of intent B of concept $\langle A, B\rangle$ into the fragment (attributes subset) of counter-example o.

If a concept avoids all such obstacles it is added to the result set of all the concepts without obstacles.

We replace a time-consuming deterministic algorithm (for instance, "Close-by-One") for generation of all concepts by the probabilistic one to randomly generate the prescribed number of concepts.

The goal of Markov chain approach is to select a random sample of formal concepts without computation of the (possibly exponential size) set $\mathbf{B}(G, M, I)$ of all the concepts.

Finally, machine learning program predicts the target class of test examples and compares the results of prediction with the original target value.

Data: random sample S of concepts, list of (τ)-objects
Result: prediction of target class of (τ)-examples
$X :=$ (τ)-examples;
for $(o \in X)$ **do**
  $PredictPositively(o) :=$ **false**;
  for $(\langle A, B\rangle \in S^+)$ **do**
    if $(B \subseteq o')$ **then**
      $PredictPositively(o) :=$ **true**;
    end
  end
end

Algorithm 3: Prediction of target class by analogy

3 Main result

Algorithm 3 gives the following

Definition 2. *Object o with fragment (attributes subset) $o' \subseteq M$ is **positively predicted** by concept $\langle A, B\rangle$ if $B \subseteq o'$.*

If there are $n = |M|$ attributes then intent B of any concept $\langle A, B\rangle$ is a point of n-hypercube $\{0, 1\}^n$.

Definition 3. ***Lower half-space*** *$H^{\downarrow}(o)$ corresponding to object o with fragment $o' \subseteq M$ is defined by linear inequality $x_{j_1} + \ldots + x_{j_k} < \frac{1}{2}$, where $M \setminus o' = \{m_{j_1}, \ldots, m_{j_k}\}$. The empty lower half-space $0 < \frac{1}{2}$ (equals to $\{0, 1\}^n$) is allowed too and corresponds to $o' = M$.*

Remark that cardinality of all possible lower half-spaces is equal to 2^n.

Key observation is

Lemma 3. *Object o is positively predicted if and only if lower half-space $H^{\downarrow}(o)$ contains a fragment B of at least one concept $\langle A, B\rangle$.*

Definition 4. *Object o is called ε-**important** if probability of occurrence of random concept $\langle A, B\rangle$ with $B \in H^{\downarrow}(o)$ is greater than ε.*

*A family of concepts is called ε-**net** if for each ε-important object o there is at least one its member $\langle A, B\rangle$ with $B \in H^{\downarrow}(o)$.*

Now we are interested only in 1-st type error probability (positive prediction fails): we need to determine a number N (depending on ε and δ) such that a random sample of cardinality N forms ε-net with probability greater than $1-\delta$.

Lemma 4. *For all ε with $l > \frac{2}{\varepsilon}$ and for any independent random samples S_1 and S_2 of concepts of cardinality l the following inequality holds:*

$$\boldsymbol{P}^{l}\{S_1 : \exists H \in (Sub\downarrow)\,[S_1 \cap H = \emptyset, \boldsymbol{P}H > \varepsilon]\} \leq \\ \leq 2 \cdot \boldsymbol{P}^{2l}\{S_1 S_2 : \exists H \in (Sub\downarrow)\,[S_1 \cap H = \emptyset, |S_2 \cap H| > \varepsilon \cdot l/2]\}.$$

Lemma 5. *For all ε and for any independent random samples S_1 and S_2 of concepts of cardinality l the following inequality holds:*

$$\boldsymbol{P}^{2l}\{S_1 S_2 : \exists H \in (Sub\downarrow)\,[S_1 \cap H = \emptyset, |S_2 \cap H| > \varepsilon \cdot l/2]\} \leq \\ \leq m^{Sub\downarrow}(2l) \cdot 2^{-\varepsilon l/2}.$$

Theorem 1. *For $n = |M|$ and for any $\varepsilon > 0$ and $1 > \delta > 0$ random sample of concepts of cardinality*

$$N \geq \frac{2 \cdot (n+1) - 2 \cdot \log_2 \delta}{\varepsilon}$$

forms ε-net with probability $> 1-\delta$.

Proof. Solve inequality $2 \cdot 2^n \cdot 2^{-\varepsilon N/2} \leq \delta$ with respect to N to obtain the estimate.

Conclusions

In this paper we provided a lower bound on sufficient number of randomly generated formal concepts to correctly predict all important positive test examples with given confidence level. The technique mostly coincides with modern approach to the famous theorem of V.N. Vapnik and A.Ya. Chervonenkis, but the situation is dual to the classical one.

Acknowledgements.

The author would like to thank Prof. Victor K. Finn and his colleagues at Federal Research Center for Computer Science and Control of Russian Academy of Science for support and helpful discussions.

The author is grateful to anonymous reviewers for improving the style of presentation.

References

1. Ganter, Bernard and Wille, Rudolf. *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, 1999
2. Kuznetsov, S.O.: A Fast Algorithm for Computing all Intersections of Objects in a Finite Semi-Lattice. Autom. Doc. Math. Linguist. 27:5, 11-21 (1993)
3. Kuznetsov, S.O. Complexity of Learning in Concept Lattices from Positive and Negative Examples. *Discrete Applied Mathematics*, no. 142(1-3). – 2004. – pp. 111–125
4. Kuznetsov, S.O. Machine Learning and Formal Concept Analysis. *Proc. 2nd International Conference on Formal Concept Analysis: Springer LNAI*, Vol. 2961. – 2004. – pp. 287-312
5. Makhalova, T.P., Kuznetsov, S.O. On Overfitting of Classifiers Making a Lattice. *Proc. 14th International Conference on Formal Concept Analysis: Springer LNAI*, Vol. 10308. – 2017. – pp. 184-197
6. Vinogradov, D.V. A Markov Chain Approach to Random Generation of Formal Concepts. *Proceedings of the Workshop Formal Concept Analysis Meets Information Retrieval (FCAIR 2013): CEUR Workshop Proceedings*, Vol. 977. – 2013. – p. 127–133
7. Vinogradov, D.V. Accidental Formal Concepts in the Presence of Counterexamples. *Proceedings of International Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD 2017): CEUR Workshop Proceedings*, Vol. 1921. – 2017. – p. 104–112
8. Vorontsov, K.V., Ivahnenko, A. Tight Combinatorial Generalization Bounds for Threshold Conjunction Rules. *Proceedings of 4th International Conference on Pattern Recognition and Machine Intelligence*. – 2011. – p. 66-73

Clustering of Biomedical Data Using the Greedy Clustering Algorithm Based on Interval Pattern Concepts*

Alexey V. Galatenko[0000−0002−7987−2738],
Stepan A. Nersisyan[0000−0002−8830−4679], and
Vera V. Pankratieva[0000−0002−9053−8319]

Faculty of Mechanics and Mathematics, Lomonosov Moscow State University,
Leninskie gory 1, 119991 Moscow, Russia

Abstract. Interval pattern concepts are a particular case of pattern structures. They can be used to clusterize rows of a numerical formal context (data matrix): two rows are close to each other if their entries at the corresponding positions fall within a given interval.
The problem of mining interval pattern concepts has much in common with the known problem related to computational geometry: given a finite set of points in the Euclidean space, position a box of a given size in such a way that it encloses as many points as possible. This problem and its variations have been thoroughly studied in the case of a plane; however, the authors are not aware of the existence of algorithms which in a reasonable time produce an exact solution in the space of an arbitrary dimension.
There exists an approximate greedy algorithm for solving this problem. It produces a solution with time which is linear in the number of points and polynomial in dimension. We apply a clustering approach based on that algorithm to the gene expression table from the dataset "The Cancer Cell Line Encyclopedia". The resulting partition well agrees with *a priori* known biological factors.

Keywords: Interval pattern concepts · Clustering · Greedy algorithms.

1 Introduction

In our days researchers frequently need to investigate various biological and medical data represented as numerical contexts (data tables). Rows of tables correspond to objects; columns correspond to attributes. It is often necessary to find clusters that are composed of objects featuring similar attributes. One of the most convenient tools that can be used for clustering this kind of data is Formal Concept Analysis.

* The research was supported by the Russian Science Foundation (project 16-11-00058 "The development of methods and algorithms for automated analysis of medical tactile information and classification of tactile images").

Formal concept analysis (FCA) is a data analysis method based on applied lattice theory and order theory. Within the framework of this theory a formal concept is defined as a pair (extent, intent) obeying the Galois connection (see the monograph [1] by B. Ganter and R. Wille).

One of the variations of FCA is known as the theory of pattern structures, which was elaborated by B. Ganter and S. Kuznetsov in [2]. An important particular case of pattern structures is interval pattern structure with the operation of interval intersection, which allows one to apply cluster analysis to rows of numerical contexts [3]. In this case similarity means that all the differences between the values of the corresponding attributes fall into given intervals.

It is easily seen that the problem of detecting similar objects can be reformulated in geometrical terms, namely, as the problem of optimal positioning of a d-dimensional box with given edge lengths for the set P of points, i.e. finding a position of the box that maximizes the number of points of the set P enclosed by the box (here $d \in \mathbb{N}$ is the number of attributes in the numerical context considered, P is the set generated by the rows of the numerical context).

In practice, biomedical data often involve thousands of entries, and each entry is described by hundreds of attributes. The existing algorithms that solve the problem of finding an optimal position of a box do not allow one to obtain an exact solution for high-dimensional data within a reasonable time. In [4] the authors introduced a fast approximate greedy algorithm for solving this problem and applied the corresponding clustering approach to the dataset of tactile images registered by the Medical Tactile Endosurgical Complex (MTEC, [5]). The experiment results demonstrated significant advantage of the proposed algorithm over the conventional k-means method in clustering quality.

In this paper we apply this clustering algorithm to the dataset "The Cancer Cell Line Encyclopedia" [6]. This dataset includes an expression table for about 20000 genes in 917 cancer cell lines. The cell lines were derived from tissues of 23 different organs. The aim of the study is to check if cancers from close organs have close gene expression values.

The rest of the paper is organised as follows. In Section 2 we introduce definitions from the formal concepts theory. In Section 3 we overview the clustering algorithm from [4]. In Section 4 we describe the procedure and present the results of application of the algorithm to the gene expression data, and in Section 5 we make concluding remarks.

2 Main Definitions

In this section we briefly recall the main definitions of the theory of formal concepts and give a geometrical interpretation of the problem of finding an interval pattern concept of maximum extent size.

Definition 1. *A semilattice operation on the partially ordered set $(M, \leq)$ is a binary operation $\sqcap\colon M \times M$ that features the following properties for a certain $e \in M$ and any elements $x, y, z \in M$:*

– $x \sqcap x = x$ *(idempotency);*
– $x \sqcap y = y \sqcap x$ *(commutativity);*
– $(x \sqcap y) \sqcap z = x \sqcap (y \sqcap z)$ *(associativity);*
– $e \sqcap x = e$.

Definition 2. *Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be partially ordered sets. A Galois connection between these sets is a pair of maps $\varphi\colon P \to Q$ and $\psi\colon Q \to P$ (each of them is referred to as a Galois operator) such that the following relations hold for any $p_1, p_2 \in P$ and $q_1, q_2 \in Q$:*

– $p_1 \leq_P p_2 \;\Rightarrow\; \varphi(p_1) \geq_Q \varphi(p_2)$ *(anti-isotone property);*
– $q_1 \leq_Q q_2 \;\Rightarrow\; \psi(q_1) \geq_P \psi(q_2)$ *(anti-isotone property);*
– $p_1 \leq_P \psi(\varphi(p_1))$ *and* $q_1 \leq_Q \varphi(\psi(q_1))$ *(isotone property).*

Applying the Galois operator twice, namely, $\psi(\varphi(p))$ and $\varphi(\psi(q))$, defines a closure operator.

Definition 3. *A closure operator $\overline{(\cdot)}$ on M is a map that assigns a closure $\overline{X} \subseteq M$ to each subset $X \subseteq M$ under the following conditions:*

– $X \leq Y \;\Rightarrow\; \overline{X} \leq \overline{Y}$ *(monotony);*
– $X \leq \overline{X}$ *(extensity);*
– $\overline{\overline{X}} = \overline{X}$ *(idempotency).*

Definition 4. *A pattern structure is a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a meet-semilattice of potential object descriptions, and $\delta\colon G \to D$ is a function that associates descriptions with objects.*

The Galois connection between the subsets of the set of objects and the set of descriptions for the pattern structure $(G, (D, \sqcap), \delta)$ is defined as follows:

$$A^{\square} := \sqcap_{g \in A} \delta(g), \qquad \text{where } A \subseteq G,$$
$$d^{\square} := \{g \in G \mid d \sqsubseteq \delta(g)\}, \qquad \text{where } A \subseteq G.$$

Definition 5. *A pattern concept of the pattern structure $(G, (D, \sqcap), \delta)$ is a pair (A, d), where $A \subseteq G$ is a subset of the set of objects and $d \in D$ is one of the descriptions in the semilattice, such that $A^{\square} = d$ and $d^{\square} = A$; A is called the pattern extent of the concept and d is the pattern intent.*

A particular case of a pattern concept is the interval pattern concept. The set D consists of rows of a numerical context which are treated as tuples of intervals of zero length. An interval pattern concept is a pair (A, d), where A is a subset of the set of objects and d is a tuple of intervals with ends determined by the smallest and the largest values of the corresponding component in the descriptions of all objects in A.

Since interval pattern concepts are determined by objects that have similarly "distributed" attributes, these concepts are convenitent to use in data clustering. The interval width can be either the same for all components (in such case it is

denoted by δ), or different for different components (in such case the widths are denoted by $\delta_1, \delta_2, \ldots, \delta_d$).

Let P be a set of n points in $\mathbb{R}^d$ ($d \in \mathbb{N}$), $\delta_1, \delta_2, \ldots, \delta_d$ be positive real numbers.

Definition 6. *A d-orthotope (also called a box) with center $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and edge lengths $\delta_1, \delta_2, \ldots, \delta_d$ is the Cartesian product of the intervals*

$$\left[x_1 - \frac{\delta_1}{2}, x_1 + \frac{\delta_1}{2}\right] \times \ldots \times \left[x_d - \frac{\delta_d}{2}, x_d + \frac{\delta_d}{2}\right].$$

It can be easily seen that the problem of identification of a maximum interval concept can be reformulated in terms of finding an *optimal* position of the box with the edge lengths $\delta_1, \delta_2, \ldots, \delta_d$, that is, maximizing the number of points of the set P enclosed by the box. This formulation can be generalized to the problem of finding an optimal position of a ball in an arbitrary metric space, since any box can be treated as a ball in the stretched L_∞ metric in which the distance $\rho(x, y)$ between the points $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ is defined as

$$\rho(x, y) = \max_{1 \le i \le d} \delta_i |x_i - y_i|.$$

3 The Greedy Clustering Algorithm Based on Interval Pattern Concepts

In this section we briefly overview the greedy clustering algorithm which was introduced in [4]. Given the set $P = \{p_i\}_{i=1}^n \subset \mathbb{R}^d$, the algorithm splits it into mutually disjoint clusters $C_1, \ldots, C_k$. The splitting procedure is based on optimal box positioning and uses a standard greedy approach. Namely, at each step an optimal position D_i of the box for the set $P \backslash (C_1, \ldots, C_{i-1})$ is determined, and C_i is assigned to be equal to $(P \setminus (C_1, \ldots, C_{i-1})) \bigcap D_i$. In order to avoid producing a big number of small clusters consisting of outliers, the algorithm uses a restriction on the number of points in the resulting clusters — they must include at least c_{min} objects. With this restriction some points can be considered unclustered.

The clustering procedure uses the approximate greedy iterative algorithm for solving the problem of an optimal box positioning. The parameters of that algorithm are the box edge lengths $\delta_1, \delta_2, \ldots, \delta_d$, the positive real numbers s, s_{min}, $\lambda < 1$ and the function $f \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. The parameters s, s_{min}, and λ regulate the duration of one iteration, while the function f returns the number of iterations for the given values n and d. Greater number of iterations and greater duration of each iteration provide better approximation.

Now we will briefly describe the greedy algorithm for finding an approximately optimal position of a box. After a short preprocessing procedure the box with the edge lengths $\delta_1, \delta_2, \ldots, \delta_d$ is transformed into the d-dimensional unit cube, and the algorithm locates the *base unit cube*, i.e. the optimal unit

cube with integer vertice coordinates. The main idea of the algorithm consists in constructing $f(n,d)$ sequences of unit cubes in such a way that each sequence starts from a random point in the base unit cube and satisfies the condition that the next cube contains more points than the previous one. After that the algorithm returns a locally optimal cube C. Each sequence is constucted iteratively. Suppose that m cubes from a sequence are already constructed. There are two possible cases.

1. If the current cube can be translated with the current step by one of the axes (the initial step size is equal to s) with an increase in the number of enclosed points, then the current cube is moved to this position.
2. Otherwise, the current step size is decreased by a factor of $\lambda < 1$. If the step size threshold $s_{min}$is reached then the procedure is terminated.

Under additional technical restrictions the authors of [4] proved the following precision and complexity bounds.

Theorem 1. *Let D_{alg} be an optimal cube produced by the algorithm and D_{opt} be a globally optimal cube. Then*

$$\frac{1}{2^d} \leq \frac{|D_{alg} \bigcap P|}{|D_{opt} \bigcap P|} \leq 1$$

and this estimate is sharp.

Theorem 2. *The algorithm for finding an approximately optimal position of the box has*

$$O\left(dn\log(n) + \frac{d^3 n^{1-\frac{1}{d}}}{s_{min}} f(n,d)\right)$$

worst-case time complexity and $O(dn)$ space complexity.

Theorem 3. *The clustering algorithm has*

$$O\left(\left(dn\log(n) + \frac{d^3 n^{1-\frac{1}{d}}}{s_{\min}} f(n,d)\right) \cdot \frac{n}{c_{\min}}\right)$$

worst-case time complexity and $O(dn)$ space complexity.

4 Applying the Clustering Algorithm to "The Cancer Cell Line Encyclopedia"

We consulted biologists and selected 432 columns of the expression table associated with genes encoding receptors, channels and transcription factors. First, we applied the clustering algorithm to the whole table. Thus, in our notation we have n equal to 917 and d equal to 432. For tuning algorithm parameters we used the following procedure. Let D denote the maximal pairwise distance

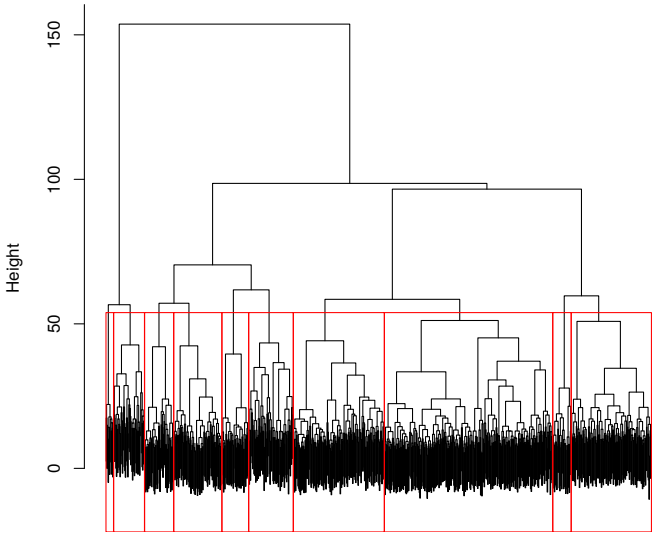


Fig. 1. Dendrogram of the hierarchial clustering of features.

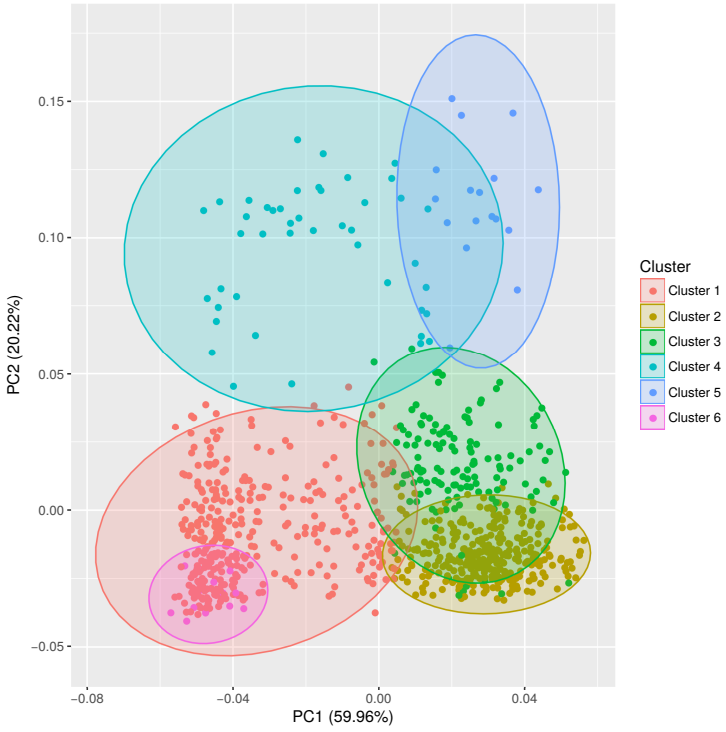


Fig. 2. Plot of the first two principal components for the clusters; 29 outlying samples are removed from this figure.

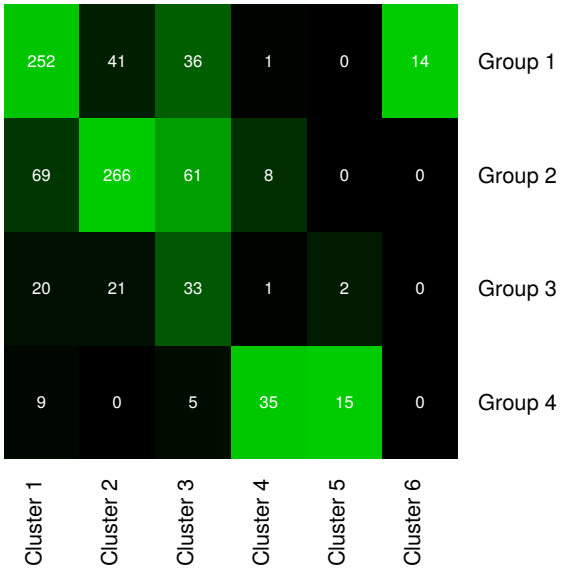


Fig. 3. Mutual arrangment of the clusters and organ groups. The number at the intersection of the ith row and the jth column indicates the number of samples which fall into both Group i and Cluster j; 29 outlying samples are excluded from consideration.

between the points considered. By the Pythagoras theorem, all points can be placed in a cube with edge length $D\sqrt{d}$. Then, a simple grid search approach on the interval $\left(0, D\sqrt{d}\right)$ was utilized for finding an acceptable cube edge length. The remaining parameters were manually tuned in order to reach acceptable (accuracy) / (running time) ratio.

We seleceted the cube edge length equal to 6.7 (i.e. $\delta_1 = \delta_2 = ... = \delta_{432} = 6.7$); c_{min}, s, s_{min} and λ were set equal to 10, 0.5, 0.3, 0.9, respectively, and the function $f(n, d)$ was taken as $[\log(dn)]$, where $[x]$ denotes the integer part of x. Despite an acceptable run time (several minutes) the results were unsatisfactory: the output of the algorithm included one huge 390-element cluster, two medium-sized 77- and 69-element clusters, and the remaining approximately optimal cubes contained less than 10 points each. This means that more than 40% of samples (381 out of 917) actually were not clusterized. Such behavior was the result of strictness of the relation "a point lies in a box" which means that each coordinate of a point must fall into a fixed range. Under this restriction, even one outlying coordinate of a point knocks it out of a cube. In high dimensional spaces single coordinate outliers are quite probable and inevitable, so before using the clustering algorithm it is reasonable to apply some dimension reduction and smoothing technique.

We applied Ward's method of hierarchial clustering to data features (R function hclust from the package stats [7] was used). The dendrogram produced (Fig. 1) was cut at height 55, which corresponds to 10 clusters. Then the expres-

sion values in the clusters were averaged. The new feature space had dimension d equal to 10. The greedy clustering algorithm was run on the dataset with reduced dimension with the cube length equal to 3; the other parameter values were left unchanged. The number of outliers essentially decreased after moving to the new agglomerated feature space — their quantity varied in the range between 25 and 35. The resulting partition consisted of 6 groups (see Fig. 2) and had an interesting biological interpretation. Namely, we calculated the number of samples in all intersections of clusters and organs. Based on this cardinalities we concluded that the clusters obtained were highly correlated with organ groups (see Fig. 3):

- Group 1: haematopoietic and lymphoid tissue, liver, skin, central nervous system, bone, soft tissue, pleura;
- Group 2: salivary gland, upper aerodigestive tract, oesophagus, biliary tract, stomach, pancreas, small intestine, large intestine, breast, thyroid, endometrium, urinary tract, lung (non-small cell cancer);
- Group 3: Kidney, ovary, prostate;
- Group 4: Autonomic ganglia and lung (small cell cancer).

It can be seen that major organ systems fall into different groups. Namely, Group 1 contains almost all non-solid organs, Group 2 contains organs from digestive system, Group 3 contains organs from genitourinary system, and Group 4 contains organs from autonomic nervous system and respiratory system. However Groups 2 and 3 seem to be dependent: Group 2 also contains some organs from genitourinary system. Thus the clusters differ by the organ systems they contain. Note that Figures 2 and 3 give ground to merge Cluster 6 with Cluster 1 and Cluster 5 with Cluster 4. The quality of the clusters can be additionally illustrated by more subtle arguments. For example, separation of small and non-small cell lung cancers seems to be reasonable, since there are some receptor coding genes which are differently expressed in these cancer types [8]. Note also that small cell lung cancer appear in the same cluster with the autonomic ganglia cancer (neuroblastoma), since their molecular mechanisms include some number of the same receptors [9].

5 Conclusions

In this paper we tested the applicability of the greedy clustering algorithm based on interval pattern concepts from the paper [4] to high-dimensional biomedical data. We showed that the clusters produced by the algorithm applied to "The Cancer Cell Line Encyclopedia" dataset were highly correlated with different organ groups and sophisticated molecular mechanismes of different cancer types.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis. Springer-Verlag, Berlin (1999)

2. Ganter, B., Kuznetsov, S.: Pattern Structures and Their Projections. Preprint MATH-AL-14-2000, Technische Universit at Dresden, Herausgeber, Der Rektor (2000)
3. Kaytoue, M., Duplessis, S., Kuznetsov, S.O., Napoli, A.: Two FCA-Based Methods for Mining Gene Expression Data. In: S. Ferré and S. Rudolph (Eds.): ICFCA 2009, LNAI 5548, pp. 2511–266, 2009.
4. Nersisyan, S.A., Pankratieva, V.V., Staroverov, V.M., Podolskii, V.E.: A Greedy Clustering Algorithm Based on Interval Pattern Concepts and the Problem of Optimal Box Positioning. Journal of Applied Mathematics, Article ID 4323590 (2017)
5. Barmin, V., Sadovnichy, ., Sokolov, M., Amiraliev, A., Pikin, O.: An original device for intraoperative detection of small indeterminate nodules. European Journal of Cardio-thoracic Surgery, **46**(6), 1027–1031 (2014)
6. Barretina, J., Caponigro, G., Stransky, N. et al.: The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature **483**(7391), 603–607 (2012)
7. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2018)
8. Wistuba, I.I., Gazdar, A.F., Minna, J.D.: Molecular genetics of small cell lung carcinoma. Seminars in Oncology **28**(2 Suppl 4), 3–13 (2001)
9. Stone, J.P., Wagner, D.D.: P-selectin mediates adhesion of platelets to neuroblastoma and small cell lung cancer. The Journal of Clinical Investigation **92**(2) (1993)

Increasing the efficiency of packet classifiers with closed descriptions

Elizaveta F. Goncharova[1][0000-0001-8358-9647] and Sergei O. Kuznetsov[2][0000-0003-3284-9001]

[1] National Research University Higher School of Economics, Moscow, Russia
{egoncharova,skuznetsov}@hse.ru

Abstract. Efficient representation of packet classifiers has become a significant challenge due to the rapid growth of data stored and processed in the forwarding, or routing, tables. In our work we propose two algorithms for reducing the size of forwarding tables both in length and width by the deletion of redundant bits and unreachable rules based on FCA analysis. We consider the task of transferring the forwarding packet to the correct destination as the task of multinomial classification. Thus, the process of reducing the forwarding table size corresponds to feature selection procedure with slight modifications. The presented techniques are based on closed descriptions and decision trees. The main challenge in applying decision trees to the task is processing the overlapping rules. To overcome this challenge we propose to employ concept-based hypotheses to delete unreachable actions assigned to the overlapping rules. The experiments were performed on data generated by the ClassBench software. The proposed approach results in significant decrease in bits in the forwarding tables as features.

Keywords: FIB optimization, concept-based hypotheses, decision tree.

1 Introduction and related works

A FIB (forwarding information base) is a wide-spread network instrument used for routing and forwarding packets to the proper output network interface. Due to the rapid growth of the forwarding tables size the time of lookup and forwarding process increases significantly. Modern networking systems require the process of packet transferring to be more and more efficient and fast. In our research we introduce a novel technique for optimization of forwarding tables. Application of the proposed algorithm results in the reduction of the number of bits, which are kept in the memory and used for the lookup process. We consider the task of transferring the forwarding packet to the correct destination in accordance with the FIB as a special task of multinomial classification, where train and test data are the same, so overfitting is not an issue. Thus, the process of reducing the size of the table is considered as the task of feature selection and rule reduction. The presented approaches are based on closed descriptions defined in terms of Formal Concept Analysis (FCA).

Some of the existing techniques for FIB optimization utilize the decision tree approach, e.g. in [1] the authors present a new algorithm using a heuristic based on the

structure built in the classifier. The main idea of algorithm *HiCut* presented in [1] is to create a decision tree based on structural properties of the classifier, where a leaf node stores just a few numbers of rules. In [2] the authors introduce a new algorithm called *HyperCut*, which is the modification of *HiCut*. Each node in the decision tree of *HyperCut* represents a *k*-dimensional hypercube. In comparison to the previous version of the algorithm the authors claim to attain 2 to 10 times memory reduction. The main problem of these approaches is processing the overlapping rules.

In [3] a novel algorithm that reveals the structural properties of FIB is proposed. The authors present a technique for reducing the number of fields (columns) in the forwarding table. The approach proposed in the article is similar to the greedy technique of feature selection. The authors are trying to reduce the rules width by selecting the fields and bits which are important for the classification process. They achieve it by sequential deletion of each field checking whether the classifier keeps the property of order-independence. We will use this algorithm as the baseline in our experiments.

The paper is organized as follows. In Section 2 we describe data and formalize the model of forwarding tables. Section 3 contains the description of the evolving approaches. Experimental results are reported in Section 4. Section 5 concludes the paper.

2 Model description

The basic scheme of packet classification using forwarding table can be presented as follows. The incoming packet goes through the table, and the first row that matches the packet description defines the respective action.

We start with the main definitions of packet forwarding. The table entry is a packet header $H = (h_1, h_2, \dots, h_n)$, $h_i \in \{0,1\}$, which is a sequence of n bits, each of them can take values zero or one. This sequence goes through the ordered set of rules $R = (r_1, r_2, \dots, r_m)$, where each rule is represented by the ordered set of m ternary values 0, 1, and * ("don't care"), and the corresponding action A_j, $j = \overline{1,N}$, where N is a number of all possible actions. This set of rules is often implemented in ternary content-addressable memory (TCAM). The forwarding process looks for exact values for all fields, assigning the packet header to the corresponding action. A header H matches a rule r_i if for every bit from H the corresponding bits from r_i takes either the same or * value [4].

The forwarding table is used with due account of the priority relation on actions. Let $P(A_i)$ be a priority of action A_i, then $P(A_i) > P(A_j)$ if $i < j$. If an input packet matches more than one rule, then the rule with the action having the highest priority is applied.

The initial packet is not given in the binary form. Packet descriptions consist of several fields, the number of fields depends on the specific protocol version (e.g., IPv4 or IPv6). In general, the source and destination hosts, port numbers, or the port numbers range make the fields of classification rule. To follow the definition mentioned above each of the field values should be performed in TCAM format. For in-

stance, the IP-address with the mask can be presented in 32-bit form, where the mask marks the significant bits. Table 1 gives an example of simplified routing table.

Table 1. Example of simplified forwarding table.

IP-address of source port	IP-address of destination port
@145.125.157.1/32	40.140.16.190/32
@195.33.215.197/32	79.205.27.10/32
@195.33.215.196/32	79.205.31.157/32

In this simplified table the rule r_i is built upon one field, which is IP-address of the source port, and the action is represented by IP-address of the destination port.

First, the initial data is transformed to TCAM format, where each number is encoded by zero, one, or * ("don't care") value. Table 2 gives an example of ternary forwarding table, where only last eight bits of the IP-address are encoded. In this example there are 7 various actions A_j and 8 features b_i, which generate the rules r_k of the forwarding table.

Table 2. An example of ternary forwarding table.

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Action
r_0	0	0	0	0	1	0	0	1	A0
r_1	1	1	0	0	0	1	0	*	A1
r_2	1	1	0	0	0	1	0	0	A2
r_3	0	0	1	1	1	1	1	1	A3
r_4	0	1	0	0	1	0	0	1	A4
r_5	0	1	0	1	1	0	1	1	A4
r_6	0	1	0	1	1	1	0	0	A5
r_7	1	0	1	1	0	0	0	1	A5
r_8	0	0	0	0	1	1	0	1	A6
r_9	0	0	0	1	1	0	0	0	A7

2.1 Data representation

The algorithms we describe below are formulated in terms of Formal Concept Analysis (FCA). To operate with TCAM data we propose a specific pattern structure [5] $(G, (D,\sqcap), \delta)$, where G is a set of objects, D is a set of all possible object descriptions, and $(D,\sqcap)$ is a meet-semi-lattice of object descriptions. Mapping $\delta: G \to D$ takes an object g to its description $d \in (D,\sqcap)$. Galois connection between $(2^G, \subseteq)$ and $(D, \sqsubseteq)$ is defined as follows [6].

$$A^{\square} = \prod_{g \in A} \delta(g), A \subseteq G,$$

$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\} \text{ for } d \in (D,\sqcap), \text{where } d \sqsubseteq \delta(g) \Leftrightarrow d \sqcap \delta(g) = d.$$

In our case $G = R$ is a set of rules, D is a set containing all possible TCAM descriptions of each rule in the alphabet $\{0, 1, *\}$, so the pattern structure is $(R, (D, \sqcap), \delta)$. The scheme of intersection operation $\sqcap$ is presented in Table 3.

Table 3. The scheme of intersection operation $\sqcap$.

$\sqcap$	0	1	*
0	0	*	*
1	*	1	*
*	*	*	*

For example, for rules $\boldsymbol{r_4}$ and $\boldsymbol{r_5}$ the result of intersection operation is the following:

$$\delta(r_{4new}) = \delta(r_4) \sqcap \delta(r_5) = \{010 * 10 * 1\}, \text{ and}$$

$$\delta(r_{4new}) \sqsubseteq \delta(r_5), \text{ as } \delta(r_{4new}) \sqcap \delta(r_5) = \delta(r_{4new}).$$

3 Optimization algorithm

We consider the general task as a standard multinomial classification problem, where the rows of the table stay for objects described by features and assigned to the corresponding classes (actions). The application of informative feature selection results in revealing the minimal combination of the informative features, thus decreasing the width of the routing table. Therefore, the look-up procedure of assigning the packet to the corresponding action can become faster. We consider two techniques based on concept-based hypotheses. The first approach is based on a variation of Close-by-One (CbO) algorithm [7]. This method results in constructing a minimal feature subset that determines the corresponding action and the reduction in the number of rules. The second approach combines concept-based hypotheses as a preprocessing step for deleting the overlapping rules with the decision tree algorithm for revealing the informative features.

3.1 Concept-based hypotheses

Concept-based hypotheses [8] used to generate rules with short premises are reformulation of JSM-hypotheses [9] in terms of formal concepts. Data can be represented by N contexts describing each of N actions (classification results) $K_i = (R_i, (D, \sqcap), \delta_i)$, $i = \overline{0, N-1}$, where R_i is a set of the i-th action examples; mapping $\delta_i: R_i \to D$ assigns an i-action example g_i to description $d \in (D, \sqcap)$, $i = \overline{0, N-1}$. The derivation operators in these contexts are defined by superscripts $\square$. Thus, the intent of i-th action examples are denoted by $r_i^{\square}$. Intents of context K_i are called i-th action intents.

3.2 Method based on Close-by-One algorithm

The first approach is based on an adaptation of CbO algorithm in the depth-first strategy [5]. The basic scheme of the proposed method is as follows.

For each context $K_i = (R_i, (D,\sqcap), \delta_i),\ i = \overline{0, N-1}$ we build the CbO tree trying to define a minimal feature subset responsible for defining the i-th action. Let R_i^{node} be a set of rules, and $R_i^{node^{\square}}$ be a common description for each rule from R_i^{node}, where $node$ is a node index in CbO tree (e.g. for the root $node$ equals zero, for the root's children nodes node will be one, etc.).

1. The root of the tree is a pair $\left(R_i^0, R_i^{0^{\square}}\right)$, where $R_i^0 = \emptyset$ and $R_i^{0^{\square}} = \emptyset$.
   Its child nodes consist of just one rule and its description $\left(R_i^1, R_i^{1^{\square}}\right)$, $R_i^1 = \{r_i^1\}$ and $R_i^{1^{\square}} = \delta(r_i^1)\ \forall r_i^1 \in R_i$. If $\delta(r_i^1)^{\square}$ includes a rule r_j^1 corresponding to action $j > i$, then the rule r_j^1 can be deleted from the routing table as unreachable. It is explained by the priority property, because each packet that satisfies r_j^1 also satisfies r_i^1, hence as $P\left(r_j^1\right) < P(r_i^1)$, r_j^1 will never be reached.
2. Having created the children nodes of the first generation, we construct the next generations of children nodes $\left(R_i^{node}, R_i^{node^{\square}}\right)$, $node > 1$. To accomplish this step we add one of the remaining rules to the previous rules set R_i^{node-1}. To get the feature-bit vector describing this new set of rules we should intersect the feature-bit vector corresponding to the added rule $\delta\left(r_i^{node}\right)$ with the current node description $R_i^{(node-1)^{\square}}$. This step can be formulated in accordance with the following rules.

$$R_i^{node} = R_i^{node-1} \cup \left\{r_i^{node}\right\},\ \forall\, r_i^{node} \in R_i \backslash R_i^{node-1};$$

$$R_i^{node^{\square}} = R_i^{node-1^{\square}} \sqcap \delta\left(r_i^{node}\right).$$

3. If $R_i^{node^{\square\square}}$ includes a rule r_j^{node} corresponding to action $j > i$, then we have got an overgeneralized description. We should return to the parent $node - 1$ and add one of the remaining rules. We aim to create the most common description of the i-th action that does not cover the description of other actions.

Example. Consider the work of the method by the example of data in Table 2. As we have only one rule for A_0, we leave it without modification, so $R_0^1 = \{r_0\}$ and $R_0^{1^{\square}} = \{00001001\}$.

The first action is also defined by one rule only: $R_1^1 = \{r_1\}$ and $R_1^{1^{\square}} = \{1100010 *\}$, however, $R_1^1 = \{r_1\}$ and $R_1^{1^{\square\square}} = \delta(r_1)^{\square} = \{r_1, r_2\}$, where r_2 defines A_2. According to the second step of the algorithm, since $P(A_1) > P(A_2)$, rule r_2 can be deleted from the routing table as unreachable.

The fourth action is determined by two rules $\{r_4, r_5\}$. The second generation of children is $R_4^2 = \{r_4, r_5\}$, $R_4^{2^{\square}} = \{010 * 10 * 1\}$. $R_4^{2^{\square\square}} = \{r_4, r_5\}$, which means that $R_4^{2^{\square}}$ is the most compact description for action A_4.

For the fifth action there are also two rules $R_5^2 = \{r_6, r_7\}$, $R_5^{2^{\square}} = \{*** 1 ** 0 *\}$. However, $R_5^{2^{\square\square}} = \{r_6, r_7, r_9\}$, where r_9 defines A_7, in accordance with the third step of

the algorithm, the obtained description $R_5^{2\square}$ is too general, and we should return to the parent nodes $R_5^1 = \{r_6\}$ and $R_5^1 = \{r_7\}$. Thus, action A_5 cannot be presented by one rule, both rules r_6 and r_7 should be kept in the final routing table.

The actions A_3, A_6 and A_7 remain the same, because they are described by one rule only. To illustrate the process described above we present the part of CbO-tree built for FIB given in Table 2 (Fig. 1). The final FIB is given in Table 4.

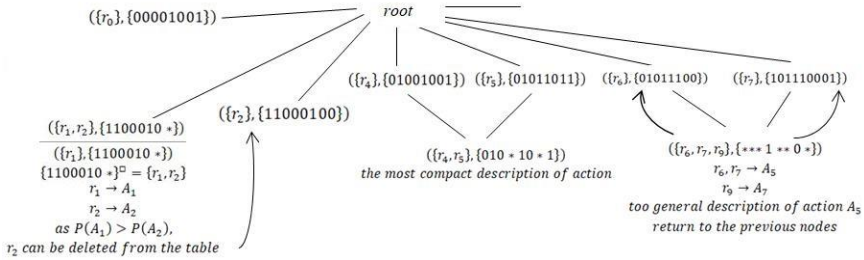


Fig 1. A part of CbO tree.

Table 4. An example of forwarding table reduced with CbO algorithm.

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Action
r_0	0	0	0	0	1	0	0	1	A0
r_1	1	1	0	0	0	1	0	*	A1
r_3	0	0	1	1	1	1	1	1	A3
r_{4new}	0	1	0	*	1	0	*	1	A4
r_6	0	1	0	1	1	1	0	0	A5
r_7	1	0	1	1	0	0	0	1	A5
r_8	0	0	0	0	1	1	0	1	A6
r_9	0	0	0	1	1	0	0	0	A7

It should be mentioned that the proposed technique does not affect the width of the routing table significantly. It can reduce the number of informative features for each action separately. Besides, it is able to decrease the length of the table by deleting unreachable actions and compressing the number of rules. In some cases the width of the table can also be reduced, for instance, if a feature-bit takes the same value for each rule in the table (i.e. the column of the table consists either of zeros, or ones), this feature-bit can be deleted from the table as uninformative.

3.3 Decision tree and concept-based hypotheses

The second approach uses concept-based hypotheses in a different way. Here we use them not for feature selection, but in the preprocessing step for deleting unreachable rules. While the final stage of feature selection is performed by a standard machine learning technique, decision tree induction in our case.

As we already mentioned some rules can be redundant in the initial table. These unreachable rules complicate the process of building the decision tree, whereas they should not be taken into account in the first way. We consider using concept-based hypotheses to detect them. Having deleted the unreachable rules we generate a decision tree. To create the optimized table we parse the decision tree finding the route for each action with zero error. A route is represented as a row from the optimized table.

As in the algorithm based on CbO we find pairs $\left(R_i^{node}, R_i^{node^{\square}}\right)$, where $R_i^{node^{\square}}$ defines the minimal description for i-th action to detect the unreachable rules. If the proportion of "*" in $R_i^{node^{\square}}$ is less than some threshold ε, and $R_i^{node^{\square\square}}$ contains a rule r_j^{node} corresponding to action $j > i$, then we perform a checking procedure as follows. For all rules $r_i^{node} \in R_i^{node^{\square\square}} \backslash \{r_j^{node}\}$ if $\delta\left(r_i^{node}\right) \sqsubseteq \delta\left(r_j^{node}\right)$, then r_j^{node} is an unreachable rule and can be deleted from the forwarding table.

Threshold ε is used to catch overgeneralized descriptions $R_i^{node^{\square}}$ that can match large number of rules, we set it to $1/2$ in this work. For instance, in example given in Section 3.2 the proportion of "*" in $R_5^{2^{\square}} = \{*** 1 ** 0 *\}$ equals to $3/4$, which is more than a half. So, we assume that it is an overgeneralized description and there is no need to compute $R_5^{2^{\square\square}}$ and check the inclusion. In this algorithm we do not aim at finding minimal hypotheses for the actions, but at deleting the unreachable rules. Thus, this stage is responsible for decreasing the length of the routing table. We should mention that ε is a hyperparameter aiming at avoiding long execution time, in our experiments the value $1/2$ has provided good performance; however, its impact could be examined more carefully in future works.

Upon deleting all unreachable rules we propose to use decision tree algorithm to find the routes that are able to distinguish all the actions. This stage results in selection of the feature-bits that are informative for classification process, this selection decreases the width of the table. The choice of decision tree algorithm is based upon two reasons:

- built-in procedure of feature selection, thus finding a rule for this or that action we obtain a short way of defining it.
- overfitting does not present a problem for this specific task, because the routing table should be an exact classifier by definition, future data cannot violate it without a general rearrangement of the routing scheme due to external reasons.

We use python implementation of decision tree classifier based on CART algorithm [10] that constructs binary tree structure and information gain for feature selection. It should be mentioned that standard machine learning techniques are not able to operate with pattern structures, therefore, to create a decision tree we encode the features with the following rules:

$$\begin{cases} b_{i0} = 1 \\ b_{i1} = 0 \end{cases}, b_i = 0; \begin{cases} b_{i0} = 0 \\ b_{i1} = 1 \end{cases}, b_i = 1; \begin{cases} b_{i0} = 0 \\ b_{i1} = 0 \end{cases}, b_i = *.$$

This encoding scheme respects the intersection operation given by Table 3. Upon processing the bits can be simply decoded into the initial ternary form.

Having created the decision tree for the initial data without the unreachable rules we can apply the simple false-positive check procedure to check the correctness of the classification results.

Thus, in the proposed method the length of the forwarding table is reduced by applying concept-based hypotheses, whereas, the decision tree with feature selection reduces the width of the table. We have applied this method to the sample FIB given in Table 2.

Example. Let us consider the optimization procedure of the sample FIB given in Table 2 using the proposed method. In this specific example the most pairs $\left(R_i^{node}, R_i^{node^{\square}}\right)$ forming the nodes of CbO tree describe one rule only (see Fig. 1), so they are not included in the procedure of unreachable rules defining. However, there are several pairs that should be processed. The first pair is $\left(R_1^1, R_1^{1^{\square}}\right) = (\{r_1\}, \{1100010 *\})$, where $R_1^{1^{\square\square}} = \{r_1, r_2\}$ and r_2 corresponds to action A_2, which has less priority than A_1 defined by r_1. Thus, we should check ε for $R_1^{1^{\square}}$; the proportion of "*" in $R_1^{1^{\square}}$ equals to $1/8$, which is less than $\varepsilon = 1/2$. This means that the description is not too general, and r_2 is a candidate for unreachable rule. Then we examine the inclusion of rules' descriptions $\delta\left(r_i^{node}\right) \sqsubseteq \delta\left(r_j^{node}\right)$. In our case

$$\delta(r_1) \sqsubseteq \delta(r_2) \Leftrightarrow \{1100010 *\} \sqcap \{11000100\} = \{1100010 *\} = \delta(r_1).$$

This means that r_2 is an unreachable rule and can be deleted from the forwarding table.

The second pair which describes more than one rule is $\left(R_4^2, R_4^{2^{\square}}\right) = (\{r_4, r_5\}, \{010 * 10 * 1\})$. The set $R_4^{2^{\square\square}} = \{r_4, r_5\}$ does not include rules corresponding to different actions (both r_4 and r_5 define action A_4) and, hence, there are no unreachable rules in this pair.

The third candidate is $\left(R_5^2, R_5^{2^{\square}}\right) = (\{r_6, r_7\}, \{*** 1 ** 0 *\})$, where $R_5^{2^{\square\square}} = \{r_6, r_7, r_9\}$. Rule r_9 corresponds to A_7, while r_6 and r_7 define A_5. However, as has been mentioned above, the proportion of "*" in $R_5^{2^{\square}}$ is greater than a threshold ε. Thus, we assume that the obtained description is too general and there are no unreachable rules in the set $R_5^{2^{\square\square}}$. In this case the assumption is correct, because action A_5 cannot be described by the one rule only, both r_6 and r_7 should be kept in the final table. Neither the description of r_6, nor the description of r_7 covers $\delta(r_9)$, which means that r_9 is a reachable rule.

Application of the preprocessing stage resulted in deleting of one unreachable rule r_2 from the initial sample table. After this step we apply decision tree procedure to generate the paths of bits, which are able to define remaining actions, and build the optimized forwarding table using these paths. The optimized version of FIB given in Table 2 is presented in Tables 5 and 6.

Table 5. An example of reduced forwarding table.

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Action
r_0	*	0	0	*	*	0	*	1	A0
r_1	*	1	*	*	0	1	*	*	A1
r_3	*	0	1	*	*	*	1	*	A3

$r_{4\text{new}}$	*	1	*	*	*	0	*	*	A4
r_6	*	1	*	*	1	1	*	*	A5
r_7	*	0	1	*	*	*	0	*	A5
r_8	*	0	0	*	*	1	*	1	A6
r_9	*	0	0	*	*	*	*	0	A7

Upon the deletion of unreachable rules the decision tree classifier has revealed the set of uninformative bits $\{b_0, b_3\}$, which are not included in any classification rule. These bits take "*" value for each rule in Table 5.

Table 6. An example of reduced forwarding table without redundant bits.

	b_1	b_2	b_4	b_5	b_6	b_7	Action
r_0	0	0	*	0	*	1	A0
r_1	1	*	0	1	*	*	A1
r_3	0	1	*	*	1	*	A3
r_4	1	*	*	0	*	*	A4
r_5	1	*	1	1	*	*	A5
r_6	0	1	*	*	0	*	A5
r_7	0	0	*	1	*	1	A6
r_8	0	0	*	*	*	0	A7

3.4 False-positive check

If we delete some bits from initial table we may have a so called false positives, when some packet satisfies the reduced table (without several bits), whereas it does not correspond to any rule in the initial FIB. To make the problem clear, let consider two reduced tables (table 4 and 6), obtained with the proposed approaches.

In accordance with the resulting table 4 the packet $h_0 = (010\mathbf{1}10\mathbf{0}1)$ will be forwarded to A_4 by r_{4new} rule, whereas h_0 does not satisfies any of the initial actions r_4 or r_5, which have been the basis for this new rule.

In table 6 the same problem occurs. For example, let $h_0 = (1\mathbf{0}\mathbf{1}011\mathbf{1}1)$ be a forwarded packet. In accordance with the values of the 1st, 2nd, and 6th bits the reduced table will assign this packet to action 3, whereas actually this packet should not be assigned to any action and should be stopped by the table.

To prevent this type of errors a false-positive check procedure should be included in the algorithms [3]. The procedure is implemented as follows, if some rules have been modified, then we should keep its initial variant in memory (32 bits and the corresponding action). Thereafter, if some input packet satisfies the new modified rule, then we check whether it also satisfies the initial rules (the ones we keep in the memory). If it suits one of them, the packet should be forwarded to the corresponding action; it is dismissed, otherwise. The process of checking is a simple comparison of two points in multidimensional space. So, the deleted bits are not included in the process of the classification procedure itself, but they are kept to prevent false positives.

4 Experimental results

The experiments were performed using the synthesized data provided with ClassBench software [11]. ClassBench generates sample routing table according to the parameters obtained from the real FIB. The synthesized tables used for the experiments consisted of the IP-address of the source port with 32-bit mask as description and IP-address of the destination port as the output action. We evaluated three generated routing tables characterized by 32 bits and consisting of 100, 500, and 906 rules, respectively.

Two proposed methods were applied to the tables described above. We compared the performance of the proposed methods with the results of the approach similar to the one presented in [3]. The authors of [3] utilize structural properties of FIB and reduce the width of the table by deleting the bits which do not affect the order-independence property. This algorithm is close to greedy technique of feature selection, where the order-independence property is checked instead of the information-gain criterion. This algorithm acts as a baseline in the experiments. The results obtained during the experiments are presented in Tables 7-9, where "Order independence" stay for the approach from [3]. We assess the performance with respect to the following properties.

Reduced number of feature-bits (column 1) shows how many bits of the 32 initial ones have been declared informative. **Reduced number of rules** (column 2) gives the amount of rules in the final table. This property demonstrates how many rules have been declared unreachable or have been united. The last property (column 3) says how many actions have been deleted from the table as unreachable.

Table 7. The results of optimization for the table with 100 rules, 32 bits, and 57 unique actions

Method	Reduced number of features	Reduced number of rules	Number of deleted actions
CbO-based	20	52	2
DT + JSM	14	59	2
Order-independence	10	86	2

Table 8. The results of optimization for the table with 500 rules, 32 bits, and 95 unique actions

Method	Reduced number of features	Reduced number of rules	Number of deleted actions
CbO-based	22	95	32
DT + JSM	15	114	32
Order-independence	29	367	32

Table 9. The results of optimization for the table with 906 rules, 32 bits, and 95 unique actions

Method	Reduced number of features	Reduced number of rules	Number of deleted actions
CbO-based	31	84	38
DT + JSM	30	79	38
Order-independence	29	309	38

We can see that the best results in reduction of table width are obtained by the decision tree algorithm in combination with concept-based hypotheses. Applying concept-based hypotheses resulted in deleting two of 57 in the first experiment, and 32 and 38 actions of 95 in the second and the third experiments respectively. This approach deleted more than a half of all initial features of the first and second synthesized FIBs. In two of three experiments the length of the table was reduced by CbO-based algorithm in the best way. It confirms the fact that the first approach succeeds in deleting redundant rules, while the other techniques are better in width reduction. It should be mentioned that we keep in the memory initial rules, which constitute the new modified rules and correspond to reachable actions, in order to perform false-positive check procedure by necessity.

The baseline approach utilizing order-independence property [3] showed the best results in minimizing the width of a forwarding table in the first experiment with 100 rules and 57 unique actions. However, it should be mentioned that respecting order-independence property one increases the number of rules. Turning the table into order-independent format requires extending of some rules and decoding the “don’t care” value into the zeros and ones in order to prevent conflict of rules.

5 Conclusion

In our work we have presented two approaches to forwarding table minimization based on decision trees and concept-based hypotheses. The first technique is based on CbO-tree construction using a special pattern structure. The second approach utilizes decision tree classification algorithm in combination with concept-based (JSM) hypotheses (DT + JSM) aiming to delete the unreachable rules and reduce the length of the table.

The experiments performed on data provided by the ClassBench software showed that the best trade-off between decreasing the width and the length of the classifier is obtained by DT + JSM technique. This method resulted in significant reduction in both the rules and bits number. The former was obtained by revealing the contradicting hypotheses and, thus, unreachable rules deletion, whereas the latter was achieved by applying the decision tree algorithm to the modified table without unreachable rules. The proposed approaches were compared to the existing technique based on keeping order-independence property of the table. Whereas the number of deleted redundant features is comparable, the number of the rules kept in the final table is

larger for the order-independent approach. The method based on CbO-tree construction resulted in significant reduction of routing table length, which was obtained by intersection of the rules corresponding to specific action; however, it could not reduce big number of features.

Overall, the proposed algorithms can be applied to the task of forwarding table minimization. In this work we overview the simplified version of the table that does not include range features. Thus, in our future research we are planning to apply interval pattern structures to process such type of fields and make our algorithms competitive with the state-of-the-art approaches.

Acknowledgements

The work of Sergei O. Kuznetsov shown in all the sections has been supported by the Russian Science Foundation grant no. 17-11-01276 and performed at St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia.

References

1. Gupta, P., McKeown, N., Classifying packets with hierarchical intelligent cuttings. Ieee Micro 20(1), 34-41(2000).
2. Singh, S., Baboescu, F., Varghese, G., Wang, J., Packet classification using multidimensional cutting. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 213-224. ACM (2003).
3. Kogan, K., Nikolenko, S. I., Rottenstreich, O., Culhane, W., Eugster, P., Exploiting order independence for scalable and expressive packet classification. IEEE/ACM Transactions on Networking, vol. 24(2), pp. 1251-1264 (2015).
4. Kogan, K., Nikolenko, S., Eugster, P., Ruan, E., Strategies for mitigating TCAM space bottlenecks. In 2014 IEEE 22nd Annual Symposium on High-Performance Interconnects pp. 25-32. IEEE (2014).
5. Ganter, B. and Kuznetsov, S., Pattern Structures and Their Projections, Proc. 9th Int. Conf. on Conceptual Structures, ICCS'01, G. Stumme and H. Delugach, Eds., Lecture Notes in Artificial Intelligence, vol. 2120, pp. 129-142 (2001).
6. Kaytoue, M., Kuznetsov, S. O., Napoli, A., and Duplessis, S., Mining gene expression data with pattern structures in formal concept analysis. Information Sciences, vol. 181(10), pp. 1989-2001 (2011).
7. Kuznetsov, S. O., Learning of simple conceptual graphs from positive and negative examples. European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, 1999, pp. 384-391.
8. Kuznetsov, S. O., Machine learning on the basis of formal concept analysis. Automation and Remote Control, vol. 62(10), pp. 1543-1564 (2001).
9. Finn, V. K., Plausible reasoning in systems of JSM type. Itogi Nauki i Tekhniki, Seriya Informatika, 1991 [in Russian].
10. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., Classification and regression trees. Belmont, CA: Wadsworth. International Group, 432 (1984).
11. ClassBench: A packet classification benchmark, http://www.arl.wustl.edu/classbench/.